Software Design Specification

Z-Wave Application Command Class Specification

Document No.:	SDS13781
Version:	15
Description:	The document describes the Z-Wave Command Classes and associated Commands used by Z-Wave enabled products at the application level.
Written By:	JFR;NOBRIOT;BBR;DEWASSIE
Date:	2020-07-06
Reviewed By:	NOBRIOT;DEWASSIE;COLSEN;JFR;JSMILJANIC
Restrictions:	Public

Approved by:

This document is the property of Silicon Labs. The data contained herein, in whole or in part, may not be duplicated, used or disclosed outside the recipient for any purpose. This restriction does not limit the recipient's right to use information contained in the data if it is obtained from another source without restriction.

REVISION RECORD				
Doc. Rev	**Date**	**By**	**Pages affected**	**Brief description of changes**
1	20170102	NOBRIOT	ALL 4.12 4.30 4.22 & 4.23 4.24 4.51 4.68	Transferred content from [2] and [3] Integrated approved content from Open Review 2016C and 2016D: • Added clarifications to Barrier Operator Command Class • Added clarifications to Configuration Command Class • Obsoleted Central Scene Command Class, version 1 and 2 • Changed and updated requirements in Central Scene Command Class, version 3 • Refactored the Irrigation Command Class, version 1 • Added the Multilevel Sensor Command Class, version 11 and moved the sensor type and scales to [17]
2	20170403	NOBRIOT	4.27, 4.29 & 4.28 4.5 & 4.21 4.115 4.22 & 4.51 4.74 4.89 4.59 4.98	Integrated approved content from Open Review 2017A: • Clarified that Color Switch supporting nodes can also use Binary Switch for brightness • Obsoleted the All Switch Command Class and the Binary Toggle Switch Command Class. • Clarified and reformatted the User Code Command Class, version 1. • Added minor editorial adjustements in the Central Scene Command Class, version 1 and the Irrigation Command Class, version 1 Integrated approved content for Notification Command Class Added Schedule Command Class, version 4 Added Meter Command Class, version 5 Moved AV Control codes to [19]
3	20170703	NOBRIOT	[15] 4.74 4.12 4.19&4.20 4.67&4.68 4.26 4.30, 4.31 & 4.32	Integrated approved content from Open Review 2017B: • Changed Time Command Class to be always present in the non-secure NIF and refactored its text • Changed Notification Command Class to allow Multi Channel Root Device to represent an aggregated binary state of all End Points • Refactored Barrier Operator Command Class • Refactored Binary Switch Command Class • Refactored Multilevel Sensor Command Class • Refactored Clock Command Class • Fixed minor text issues in Configuration Command Class
4	20171002	NOBRIOT	4.31 4.37 3.4 4.113 & 4.114 4.98 4.83, 4.84 & 4.85	Integrated approved content from Open Review 2017C: • Clarified Configuration Command Class: parameters in bulk commands must be in a consecutive range • Clarified Door Lock Command Class: a supporting node must apply constant mode even if operating in Timed Operation • Updated NIF section to reflect its contents based on Security bootstrapping event. • Clarified Thermostat Setpoint Command Class and what to return in a report when a non-supported type is requested. • Obsoleted the proprietary Item ID fields in the Simple AV Control Command Class • Refactored Scene Activation, Scene Actuator Configuration and Scene Controller Configuration Command Classes.

REVISION RECORD				
Doc. Rev	**Date**	**By**	**Pages affected**	**Brief description of changes**
5	20180110	NOBRIOT	4.27 4.99 4.33 4.74	Integrated approved content from Open Review 2017D: • Obsoleted indexed colors in the Color Switch Command Class • Changed the allowed value range for the Sound Switch Command Class • Clarified Configuration Command “re-inclusion required” parameters • Added an explicit statement in Notification Command Class stating that Pull Sensor are not recommended
6	20180305	BBR	All	Added Silicon Labs template
7	20180409	NOBRIOT	4.39 4.64 4.33 4.55, 4.56, 4.57, 4.58, & 4.59 4.62, 4.63 & 4.64	Integrated approved content from Open Review 2018A: New Command Classes: • Added Door Lock Command Class, version 4 • Added the Meter Table Monitor, version 3. Minor changes: • Moved the Notification Command Class Controller guidelines to [23] • Renamed configuration parameters “re-inclusion required” parameters to “Altering capabilities” parameters Rewriting with RFC keywords: • Meter Command Class, version 1-5 • Meter Table Monitor, version 1-2
8	20180701	NOBRIOT	[15] 4.47 4.49 4.116 4.108 & 4.109 4.113 & 4.114	Integrated content from 2018B Contributions: • Moved the Time and Time Parameters Command Classes in the Management Command Class specification. New Command Classes: • Added a version 2 of the Humidity Control Mode Command Class to support an auto mode • Added a version 2 of the Humidity Control Setpoint Command Class to support an auto mode. • Added the User Code Command Class, version 2. Rewriting with RFC keywords: • Thermostat Mode Command Class, version 1-3 • Thermostat Setpoint Command Class, version 1-3
9	20181001	NOBRIOT	3.6 4.24.7 4.115 & 4.116	2018C contributions: • Updated the list of Actuator Command Classes • Relaxed requirement level for refresh rate with Central Scene Command Class • Clarified the use of User Identifier value 0 in the User Code Command Class
10	20190101	NOBRIOT, DEWASSIE	3 4.116 4.117 4.113.1	2018D Contributions: • Added requirement numbers for generic command class section. • Fixed bitmask length field in User Code Command Class, version 2 • Fixed Window Covering inconsistencies • Updated the Thermostat Setpoint Command Class, version 1-2 to be mandatory compliant with Interpretation A.

REVISION RECORD				
Doc. Rev	**Date**	**By**	**Pages affected**	**Brief description of changes**
11	20190401	NORBIOT DEWASSIE	4.115 4.50 4.10 4.11 4.106	2019A Contributions: • Clarified field usage in the User Code Command Class • Added IR Repeater Command Class • Added Authentication Command Class • Added Authentication Media Command Class • Added Thermostat Fan Mode Command Class, version 5
12	20190701	NOBRIOT	4.68.2 4.117.3 4.39 4.100 4.43 4.47	Contributions 2019B: • Clarified that returning an answer to Multilevel Sensor Get Supported Scale Command is mandatory • Reorganized text about odd/event parameter IDs in the Window Covering Command Class • Clarified bolt/latch definitions and door component encoding in the Door Lock Command Class • Added the Sound Switch Command Class, version 2. • Added the Generic Schedule Command Class, version 1. • Fixed a typo in the list of available modes for Humidity Control Mode Command Class, version 2
13	20191001	DEWASSIE	4.10.6 4.10.9 4.116.16 4.24 4.60 4.117.7	Contributions 2019C: • Added a clarification in Report more field in Authentication Data Get Command • Added a clarification in Report more field in Authentication Technology Combination Get Command • Added a clarification in Report more field in Extended User Code Get Command • Added a requirement for a node supporting Central Scene Command Class • Added Meter Command Class, version 6 • Clarified window target value requirements under Window Covering Report Command
14	20200101	NOBRIOT DEWASSIE	3.4.3.1 4.19 4.7 4.8 4.9 4.99 4.24, 4.26 4.30 & 4.44	Contributions 2019D: • Added examples under the NIF and its contents • Updated the Value interpretation in the Binary Switch Command Class • Deprecated the Anti-Theft Command Class, version 2 • Added Anti-Theft Command Class, version 3 • Added Anthi-Theft Unlock Command Class, version 1 • Clarified how to encode volume in the Sound Switch Play Report • Added a Multi Channel Consideration section for Command Classes that should only be supported at the root device
15	20200701	DEWASSIE NOBRIOT	4.10.8 4.116.15 4.100.3 4.39.1 4.74.2.11 Table 40 4.60.2	Contibutions 2020A: • Enabled the Authentication Technology Combination Set command to schedule a using only User Code. • Relaxed the Master Code duplicate detection requirement • Clarified a field dependency in the Sound Switch Tone Play Report Command • Clarified that Auto-Relock for Door Locks should not have any effect in Timed Operation • Removed a table reference error in Notification Command Class • Fixed maximum lock timeout in minute value. • Clarified what scale to apply when a supporting node reset the meter value.

Table of Contents

1 ABBREVIATIONS 21

2 INTRODUCTION 22

2.1 Precedence of definitions 22
2.2 Terms used in this document 22

3 COMMAND CLASS OVERVIEW 22

3.1 Overview 23
3.2 Command class format 23
3.2.1 Frame format 23
3.2.2 Command class versioning 25
3.3 Controlled and Supported Command Classes 25
3.4 Node Information Frame 25
3.4.1 Z-Wave Protocol Specific Part 26
3.4.2 Application Specific Part 27
3.4.3 NIF and Multi Channel/Security Command Classes 28
3.4.4 Command Class specific NIF rules 31
3.5 Multicast and broadcast commands 31
3.6 Actuator Control 31
3.6.1 Terminology 32
3.6.2 Reporting values 32
3.6.3 Command values vs. hardware values 33
3.6.4 Supporting multiple actuator Command Classes 33
3.7 Common fields and encoding 34
3.7.1 Reserved and Res fields 34
3.7.2 Reserved values and reserved bits 34
3.7.3 Duration encoding 34
3.7.4 Unsigned encoding 35
3.7.5 Signed encoding 35
3.7.6 Fields values and version 35

4 COMMAND CLASS DEFINITIONS 36

4.1 Alarm Command Class, version 1 [DEPRECATED] 37
4.1.1 Interoperability considerations 37
4.1.2 Alarm Get Command 37
4.1.3 Alarm Report Command 38
4.2 Alarm Command Class, version 2 [DEPRECATED] 39
4.2.1 Interoperability considerations 39
4.2.2 Alarm Set Command 39
4.2.3 Alarm Get Command 40
4.2.4 Alarm Report Command 41
4.2.5 Alarm Type Supported Get Command 42
4.2.6 Alarm Type Supported Report Command 43
4.3 Alarm Sensor Command Class, version 1 [DEPRECATED] 44

4.3.1 Alarm Sensor Get Command 44
4.3.2 Alarm Sensor Report Command 45
4.3.3 Alarm Sensor Supported Get Command 46
4.3.4 Alarm Sensor Supported Report Command 46
4.4 Alarm Silence Command Class, version 1 48
4.4.1 Alarm Silence Set Command 48
4.5 All Switch Command Class, version 1 [OBSOLETED] 50
4.5.1 All Switch Set Command 50
4.5.2 All Switch Get Command 51
4.5.3 All Switch Report Command 51
4.5.4 All Switch On Command 51
4.5.5 All Switch Off Command 52
4.6 Anti-theft Command Class, version 1 [OBSOLETED] 53
4.7 Anti-theft Command Class, version 2 [DEPRECATED] 54
4.7.1 Anti-theft Set Command 55
4.7.2 Anti-theft Get Command 56
4.7.3 Anti-theft Report Command 57
4.7.4 Examples 59
4.8 Anti-theft Command Class, version 3 61
4.8.1 Compatibility Considerations 61
4.8.2 Anti-Theft Set Command 62
4.8.3 Anti-theft Get Command 64
4.8.4 Anti-Theft Report Command 64
4.9 Anti-theft Unlock Command Class, version 1 67
4.9.1 Compatibility Considerations 67
4.9.2 Anti-Theft Unlock State Get Command 67
4.9.3 Anti-Theft Unlock State Report Command 68
4.9.4 Anti-Theft Unlock Set Command 70
4.10 Authentication Command Class, version 1 71
4.10.1 Terminology 71
4.10.2 Interoperability Considerations 71
4.10.3 Authentication Capability Get Command 72
4.10.4 Authentication Capability Report Command 72
4.10.5 Authentication Data Set Command 76
4.10.6 Authentication Data Get Command 77
4.10.7 Authentication Data Report Command 79
4.10.8 Authentication Technologies Combination Set Command 81
4.10.9 Authentication Technologies Combination Get Command 85
4.10.10 Authentication Technologies Combination Report Command 87
4.10.11 Authentication Checksum Get Command 91
4.10.12 Authentication Checksum Report Command 92
4.10.13 Examples and use-cases 95
4.11 Authentication Media Write Command Class, version 1 97
4.11.1 Authentication Media Capability Get Command 97
4.11.2 Authentication Media Capability Report Command 97
4.11.3 Authentication Media Write Start Command 98
4.11.4 Authentication Media Write Stop Command 99

4.11.5 Authentication Media Write Status Command 100
4.12 Barrier Operator Command Class, version 1 102
4.12.1 Compatibility considerations 102
4.12.2 Barrier Operator Set Command 102
4.12.3 Barrier Operator Get Command 104
4.12.4 Barrier Operator Report Command 104
4.12.5 Barrier Operator Get Signaling Capabilities Supported Command 105
4.12.6 Barrier Operator Report Signaling Capabilities Supported Command 105
4.12.7 Barrier Operator Event Signal Set Command 106
4.12.8 Barrier Operator Event Signaling Get Command 107
4.12.9 Barrier Operator Event Signaling Report 108
4.13 Basic Command Class, version 1 109
4.13.1 Compatibility considerations 109
4.13.2 Basic Set Command 109
4.13.3 Basic Get Command 110
4.13.4 Basic Report Command 110
4.14 Basic Command Class, version 2 111
4.14.1 Compatibility considerations 111
4.14.2 Basic Report Command 111
4.15 Basic Tariff Information Command Class, version 1 113
4.15.1 Basic Tariff Information Get Command 113
4.15.2 Basic Tariff Information Report Command 114
4.16 Basic Window Covering Command Class, version 1 [OBSOLETED] 116
4.16.1 Basic Window Covering Start Level Change Command 116
4.16.2 Basic Window Covering Stop Level Change Command 116
4.17 Binary Sensor Command Class, version 1 [DEPRECATED] 117
4.17.1 Binary Sensor Get Command 117
4.17.2 Binary Sensor Report Command 117
4.18 Binary Sensor Command Class, version 2 [DEPRECATED] 118
4.18.1 Binary Sensor Get Command 118
4.18.2 Binary Sensor Report Command 119
4.18.3 Binary Sensor Get Supported Sensor Command 120
4.18.4 Binary Sensor Supported Sensor Report Command 120
4.19 Binary Switch Command Class, version 1 121
1.1.1 Binary Switch Set Command 121
1.1.2 Binary Switch Get Command 122
1.1.3 Binary Switch Report Command 122
4.20 Binary Switch Command Class, version 2 123
4.20.1 Compatibility considerations 123
4.20.2 Binary Switch Set Command 123
4.20.3 Binary Switch Report Command 124
4.21 Binary Toggle Switch Command Class, version 1 [OBSOLETED] 125
4.21.1 Binary Toggle Switch Set Command 125
4.21.2 Binary Toggle Switch Get Command 125
4.21.3 Binary Toggle Switch Report Command 125
4.22 Central Scene Command Class, version 1 [OBSOLETED] 127
4.22.1 Central Scene Supported Get Command 128

4.22.2 Central Scene Supported Report Command 128
4.22.3 Central Scene Notification Command 129
4.23 Central Scene Command Class, version 2 [OBSOLETED] 131
4.23.1 Compatibility considerations 131
4.23.2 Central Scene Supported Report Command 131
4.23.3 Central Scene Notification Command 133
4.24 Central Scene Command Class, version 3 134
4.24.1 Compatibility considerations 135
4.24.2 Central Scene Configuration Set Command 136
4.24.3 Central Scene Configuration Get Command 137
4.24.4 Central Scene Configuration Report Command 137
4.24.5 Central Scene Supported Get Command 138
4.24.6 Central Scene Supported Report Command 138
4.24.7 Central Scene Notification Command 139
4.25 Climate Control Schedule Command Class, version 1 [DEPRECATED] 140
4.25.1 Schedule Set Command 142
4.25.2 Schedule Get Command 143
4.25.3 Schedule Report Command 144
4.25.4 Schedule Changed Get Command 144
4.25.5 Schedule Changed Report Command 144
4.25.6 Schedule Override Set Command 145
4.25.7 Schedule Override Get Command 147
4.25.8 Schedule Override Report Command 147
4.26 Clock Command Class, version 1 147
1.1.1 Interoperability considerations 147
4.26.1 Multi Channel Considerations 148
4.26.2 Clock Set Command 148
1.1.2 Clock Get Command 149
1.1.3 Clock Report Command 150
4.27 Color Switch Command Class, version 1 151
4.27.1 Compatibility considerations 151
4.27.2 Interoperability considerations 151
4.27.3 Color Switch Supported Get Command 152
4.27.4 Color Switch Supported Report Command 152
4.27.5 Color Switch Get Command 153
4.27.6 Color Switch Report Command 153
4.27.7 Color Switch Set Command 154
4.27.8 Color Switch Start Level Change Command 156
4.27.9 Color Switch Stop Level Change Command 157
4.28 Color Switch Command Class, version 2 158
4.28.1 Compatibility considerations 158
4.28.2 Interoperability considerations 158
4.28.3 Color Switch Set Command 158
4.29 Color Switch Command Class, version 3 160
4.29.1 Compatibility considerations 160
4.29.2 Interoperability considerations 160
4.29.3 Color Switch Report Command 160

4.29.4 Color Switch Start Level Change Command 161
4.30 Configuration Command Class, version 1 163
4.30.1 Compatibility considerations 163
4.30.2 Configuration Set Command 164
4.30.3 Configuration Get Command 165
4.30.4 Configuration Report Command 166
4.31 Configuration Command Class, version 2 167
4.31.1 Compatibility considerations 167
4.31.2 Interoperability considerations 167
4.31.3 Configuration Set Command 168
4.31.4 Configuration Bulk Set Command 169
4.31.5 Configuration Bulk Get Command 171
4.31.6 Configuration Bulk Report Command 172
4.32 Configuration Command Class, version 3 174
4.32.1 Compatibility considerations 174
4.32.2 Interoperability considerations 175
4.32.3 Configuration Name Get Command 175
4.32.4 Configuration Name Report Command 176
4.32.5 Configuration Info Get Command 177
4.32.6 Configuration Info Report Command 178
4.32.7 Configuration Properties Get Command 179
4.32.8 Configuration Properties Report Command 180
4.33 Configuration Command Class, version 4 183
4.33.1 Compatibility considerations 183
4.33.2 Configuration Set Command 185
4.33.3 Configuration Bulk Set Command 186
4.33.4 Configuration Properties Report Command 187
4.33.5 Configuration Default Reset Command 188
4.34 Controller Replication Command Class, version 1 189
4.34.1 Transfer Group Command 189
4.34.2 Transfer Group Name Command 189
4.34.3 Transfer Scene Command 190
4.34.4 Transfer Scene Name Command 191
4.35 Demand Control Plan Configuration Command Class, version 1 192
4.35.1 DCP List Supported Get Command 192
4.35.2 DCP List Supported Report Command 193
4.35.3 DCP List Set Command 194
4.35.4 DCP List Remove 199
4.36 Demand Control Plan Monitor Command Class, version 1 200
4.36.1 DCP List Get Command 200
4.36.2 DCP List Report Command 201
4.36.3 DCP Event Status Get 202
4.36.4 DCP Event Status Report 203
4.37 Door Lock Command Class, version 1-2 204
4.37.1 Compatibility considerations 204
4.37.2 Door Lock Operation Set Command 204
4.37.3 Door Lock Operation Get Command 205

4.37.4 Door Lock Operation Report Command 205
4.37.5 Door Lock Configuration Set Command 208
4.37.6 Door Lock Configuration Get Command 210
4.37.7 Door Lock Configuration Report Command 211
4.38 Door Lock Command Class, version 3 212
4.38.1 Compatibility considerations 212
4.38.2 Door Lock Operation Report Command 212
4.39 Door Lock Command Class, version 4 213
4.39.1 Terminology 213
1.1.1 Compatibility considerations 214
4.39.2 Door Lock Operation Set Command 215
4.39.3 Door Lock Operation Get Command 216
4.39.4 Door Lock Operation Report Command 216
4.39.5 Door Lock Configuration Set Command 220
4.39.6 Door Lock Configuration Get Command 223
4.39.7 Door Lock Configuration Report Command 223
1.1.2 Door Lock Capabilities Get Command 224
1.1.3 Door Lock Capabilities Report Command 224
4.40 Door Lock Logging Command Class, version 1 228
4.40.1 Door Lock Logging Records Supported Get Command 228
4.40.2 Door Lock Logging Records Supported Report Command 228
4.40.3 Door Lock Logging Record Get Command 229
4.40.4 Door Lock Logging Record Report Command 230
4.41 Energy Production Command Class, version 1 233
4.41.1 Energy Production Get Command 233
4.41.2 Energy Production Report Command 234
4.42 Entry Control Command Class, version 1 235
4.42.1 Interoperability Considerations 235
4.42.2 Security Considerations 235
4.42.3 Handling user supplied data 236
4.42.4 Handling Incorrect Entry 236
4.42.5 Entry Control Notification Command 237
4.42.6 Entry Control Key Supported Get Command 239
4.42.7 Entry Control Key Supported Report Command 239
4.42.8 Entry Control Event Supported Get Command 240
4.42.9 Entry Control Event Supported Report Command 241
4.42.10 Entry Control Configuration Set Command 242
4.42.11 Entry Control Configuration Get Command 243
4.42.12 Entry Control Configuration Report Command 243
4.42.13 Event Types 244
4.43 Generic Schedule Command Class, version 1 246
4.43.1 Terminology 246
4.43.2 Interoperability considerations 246
1.1.1 Generic Schedule Capabilities Get Command 246
1.1.2 Generic Schedule Capabilities Report Command 247
1.1.3 Generic Schedule Time Range Set Command 249
1.1.4 Generic Schedule Time Range Get Command 255

1.1.5 Generic Schedule Time Range Report Command 256
1.1.6 Generic Schedule Set Command 257
1.1.7 Generic Schedule Get Command 260
1.1.8 Generic Schedule Report Command 261
4.44 Geographic Location Command Class, version 1 263
4.44.1 Multi Channel Considerations 263
4.44.2 Geographic Location Set Command 263
4.44.3 Geographic Location Get Command 264
4.44.4 Geographic Location Report Command 264
4.45 HRV Status Command Class, version 1 265
4.45.1 HRV Status Get Command 265
4.45.2 HRV Status Report Command 266
4.45.3 HRV Status Supported Get Command 267
4.45.4 HRV Status Supported Report Command 267
4.46 HRV Control Command Class, version 1 269
4.46.1 HRV Mode Set 269
4.46.2 HRV Mode Get Command 270
4.46.3 HRV Mode Report Command 270
4.46.4 HRV Bypass Set Command 270
4.46.5 HRV Bypass Get Command 271
4.46.6 HRV Bypass Report Command 271
4.46.7 HRV Ventilation Rate Set Command 272
4.46.8 HRV Ventilation Rate Get Command 272
4.46.9 HRV Ventilation Rate Report Command 272
4.46.10 HRV Mode Supported Get Command 273
4.46.11 HRV Mode Supported Report Command 273
4.47 Humidity Control Mode Command Class, version 1-2 275
4.47.1 Humidity Control Mode Set Command 275
4.47.2 Humidity Control Mode Get Command 276
4.47.3 Humidity Control Mode Report Command 276
4.47.4 Humidity Control Mode Supported Get Command 277
4.47.5 Humidity Control Mode Supported Report Command 277
4.48 Humidity Control Operating State Command Class, version 1 279
4.48.1 Humidity Control Operating State Get Command 279
4.48.2 Humidity Control Operating State Report Command 279
4.49 Humidity Control Setpoint Command Class, version 1-2 281
4.49.1 Humidity Control Setpoint Set Command 281
4.49.2 Humidity Control Setpoint Get Command 283
4.49.3 Humidity Control Setpoint Report Command 283
4.49.4 Humidity Control Setpoint Supported Get Command 284
4.49.5 Humidity Control Setpoint Supported Report Command 284
4.49.6 Humidity Control Setpoint Scale Supported Get Command 286
4.49.7 Humidity Control Setpoint Scale Supported Report Command 286
4.49.8 Humidity Control Setpoint Capabilities Get Command 287
4.49.9 Humidity Control Setpoint Capabilities Report Command 288
4.50 IR Repeater Command Class, version 1 289
1.1.1 IR Repeater Capabilities Get Command 289

1.1.2 IR Repeater Capabilities Report Command 289
1.1.3 IR Repeater IR Code Learning Start Command 293
1.1.4 IR Repeater IR Code Learning Stop Command 294
1.1.5 IR Repeater IR Code Learning Status Command 294
1.1.6 IR Repeater Learnt IR Code Remove Command 296
1.1.7 IR Repeater Learnt IR Code Get Command 296
1.1.8 IR Repeater Learnt IR Code Report Command 297
1.1.9 IR Repeater Learnt IR Code Readback Get Command 298
1.1.10 IR Repeater Learnt IR Code Readback Report Command 299
1.1.11 IR Repeater Configuration Set Command 300
1.1.12 IR Repeater Configuration Get Command 303
1.1.13 IR Repeater Configuration Report Command 303
1.1.14 IR Repeater Repeat Learnt Code Command 304
1.1.15 IR Repeater Repeat Command 305
4.51 Irrigation Command Class, version 1 309
4.51.1 Terminology 309
4.51.2 Compatibility considerations 309
4.51.3 Interoperability considerations 309
4.51.4 Irrigation System Info Get Command 311
4.51.5 Irrigation System Info Report Command 311
4.51.6 Irrigation System Status Get Command 312
4.51.7 Irrigation System Status Report Command 313
4.51.8 Irrigation System Config Set Command 317
4.51.9 Irrigation System Config Get Command 319
4.51.10 Irrigation System Config Report Command 319
4.51.11 Irrigation Valve Info Get Command 320
4.51.12 Irrigation Valve Info Report Command 321
4.51.13 Irrigation Valve Config Set Command 323
4.51.14 Irrigation Valve Config Get Command 326
4.51.15 Irrigation Valve Config Report Command 327
4.51.16 Irrigation Valve Run Command 328
4.51.17 Irrigation Valve Table Set Command 329
4.51.18 Irrigation Valve Table Get Command 330
4.51.19 Irrigation Valve Table Report Command 330
4.51.20 Irrigation Valve Table Run Command 331
4.51.21 Irrigation System Shutoff Command 332
4.52 Language Command Class, version 1 333
4.52.1 Language Set Command 333
4.52.2 Language Get Command 334
4.52.3 Language Report Command 335
4.53 Lock Command Class, version 1 [DEPRECATED] 336
4.53.1 Lock Set Command 336
4.53.2 Lock Get Command 336
4.53.3 Lock Report Command 337
4.54 Manufacturer Proprietary Command Class, version 1 338
4.54.1 Manufacturer Proprietary Command 338
4.55 Meter Command Class, version 1 339

4.55.1 Terminology 339
1.1.1 Meter Get Command 340
1.1.2 Meter Report Command 340
4.56 Meter Command Class, version 2 341
4.56.1 Compatibility Considerations 341
4.56.2 Meter Supported Get Command 342
4.56.3 Meter Supported Report Command 342
4.56.4 Meter Reset Command 343
4.56.5 Meter Get Command 343
4.56.6 Meter Report Command 344
4.57 Meter Command Class, version 3 346
4.57.1 Compatibility Considerations 346
4.57.2 Meter Supported Report Command 347
4.57.3 Meter Get Command 347
4.57.4 Meter Report Command 348
4.58 Meter Command Class, version 4 349
4.58.1 Compatibility Considerations 349
4.58.2 Meter Supported Report Command 349
4.58.3 Meter Get Command 351
4.58.4 Meter Report Command 353
4.59 Meter Command Class, version 5 355
4.59.1 Compatibility considerations 355
4.59.2 Meter Supported Report Command 355
4.59.3 Meter Report Command 357
4.60 Meter Command Class, version 6 359
4.60.1 Compatibility Considerations 359
4.60.2 Meter Reset Command 359
4.61 Meter Table Configuration Command Class, version 1 360
4.61.1 Meter Table Point Adm Number Set Command 360
4.62 Meter Table Monitor Command Class, version 1 361
4.62.1 Meter Table Point Adm. Number Get Command 361
4.62.2 Meter Table Point Adm. Number Report Command 361
4.62.3 Meter Table ID Get Command 362
4.62.4 Meter Table ID Report Command 362
4.62.5 Meter Table Capability Get Command 363
4.62.6 Meter Table Capability Report Command 364
4.62.7 Meter Table Status Supported Get Command 365
4.62.8 Meter Table Status Supported Report Command 366
4.62.9 Meter Table Status Depth Get Command 367
4.62.10 Meter Table Status Date Get Command 368
4.62.11 Meter Table Status Report Command 370
4.62.12 Meter Table Current Data Get Command 372
4.62.13 Meter Table Current Data Report Command 373
4.62.14 Meter Table Historical Data Get Command 376
4.62.15 Meter Table Historical Data Report Command 378
4.63 Meter Table Monitor Command Class, version 2 380
4.63.1 Compatibility Considerations 381

4.63.2 Meter Table Point Adm. Number Report Command 381
4.63.3 Meter Table ID Report Command 381
4.63.4 Meter Table Capability Report Command 382
4.63.5 Meter Table Current Data Report Command 383
4.63.6 Meter Table Historical Data Report Command 385
4.64 Meter Table Monitor Command Class, version 3 386
4.64.1 Compatibility Considerations 386
4.65 Meter Table Push Configuration Command Class version 1 386
4.65.1 Meter Table Push Configuration Set Command 387
4.65.2 Meter Table Push Configuration Get Command 388
4.65.3 Meter Table Push Configuration Report Command 389
4.66 Move To Position Window Covering Command Class, version 1 [OBSOLETED] 389
4.66.1 Move To Position Set Command 390
4.66.2 Move To Position Get Command 390
4.66.3 Move To Position Report Command 391
4.67 Multilevel Sensor Command Class, version 1-4 391
4.67.1 Multilevel Sensor Get Command 391
4.67.2 Multilevel Sensor Report Command 391
4.68 Multilevel Sensor Command Class, Version 5-11 393
4.68.1 Compatibility considerations 393
4.68.2 Multilevel Sensor Get Supported Sensor Command 394
4.68.3 Multilevel Sensor Supported Sensor Report Command 394
4.68.4 Multilevel Sensor Get Supported Scale Command 395
4.68.5 Multilevel Sensor Supported Scale Report Command 396
4.68.6 Multilevel Sensor Get Command 397
4.68.7 Multilevel Sensor Report Command 398
4.69 Multilevel Switch Command Class, version 1 403
4.69.1 Multilevel Switch Set Command 404
4.69.2 Multilevel Switch Get Command 405
4.69.3 Multilevel Switch Report Command 405
4.69.4 Multilevel Switch Start Level Change Command 406
4.69.5 Multilevel Switch Stop Level Change Command 407
4.70 Multilevel Switch Command Class, version 2 407
4.70.1 Compatibility considerations 407
4.70.2 Multilevel Switch Set Command 407
4.70.3 Multilevel Switch Start Level Change Command 408
4.71 Multilevel Switch Command Class, version 3 409
4.71.1 Compatibility considerations 409
4.71.2 Multilevel Switch Supported Get Command 410
4.71.3 Multilevel Switch Supported Report Command 410
4.71.4 Multilevel Switch Start Level Change Command 412
4.72 Multilevel Switch Command Class, version 4 414
4.72.1 Compatibility considerations 414
4.72.2 Multilevel Switch Report Command 415
4.73 Multilevel Toggle Switch Command Class, version 1 [DEPRECATED] 416
4.73.1 Multilevel Toggle Switch Set Command 416
4.73.2 Multilevel Toggle Switch Get Command 416

4.73.3 Multilevel Toggle Switch Report Command 417
4.73.4 Multilevel Toggle Switch Start Level Change Command 417
4.73.5 Multilevel Toggle Switch Stop Level Change Command 418
4.74 Notification Command Class, version 3-8 418
4.74.1 Terminology 418
4.74.2 Compatibility considerations 420
4.74.3 Interoperability considerations 428
4.74.4 Notification Set Command 429
4.74.5 Notification Get Command 430
4.74.6 Notification Report Command 432
4.74.7 Notification Supported Get Command 437
4.74.8 Notification Supported Report Command 437
4.74.9 Event Supported Get Command 438
4.74.10 Event Supported Report Command 439
4.75 Prepayment Command Class, version 1 440
4.75.1 Prepayment Balance Get Command 440
4.75.2 Prepayment Balance Report Command 441
4.75.3 Prepayment Supported Get Command 443
4.75.4 Prepayment Supported Report Command 443
4.76 Prepayment Encapsulation Command Class, version 1 444
4.76.1 Prepayment Encapsulation Command 444
4.77 Proprietary Command Class, version 1 [DEPRECATED] 444
4.77.1 Proprietary Set Command 445
4.77.2 Proprietary Get Command 445
4.77.3 Proprietary Report Command 446
4.78 Protection Command Class, version 1 446
4.78.1 Protection Set Command 446
4.78.2 Protection Get Command 447
4.78.3 Protection Report Command 448
4.79 Protection Command Class, version 2 448
4.79.1 Protection Set Command 448
4.79.2 Protection Report Command 450
4.79.3 Protection Supported Get Command 450
4.79.4 Protection Supported Report Command 450
4.79.5 Protection Exclusive Control 451
4.79.6 Protection Timeout 452
4.80 Pulse Meter Command Class, version 1 [DEPRECATED] 454
4.80.1 Pulse Meter Get Command 454
4.80.2 Pulse Meter Report Command 455
4.81 Rate Table Configuration Command Class, version 1 456
4.81.1 Rate Table Set Command 456
4.81.2 Rate Table Remove Command 459
4.82 Rate Table Monitor Command Class, version 1 460
4.82.1 Rate Table Supported Get Command 460
4.82.2 Rate Table Supported Report Command 460
4.82.3 Rate Table Get Command 462
4.82.4 Rate Table Report Command 462

4.82.5 Rate Table Active Rate Get Command 463
4.82.6 Rate Table Active Rate Report Command 463
4.82.7 Rate Table Current Data Get Command 464
4.82.8 Rate Table Current Data Report Command 465
4.82.9 Rate Table Historical Data Get Command 467
4.82.10 Rate Table Historical Data Report Command 469
4.83 Scene Activation Command Class, version 1 471
4.83.1 Compatibility considerations 471
4.83.2 Scene Activation Set Command 471
4.84 Scene Actuator Configuration Command Class, version 1 472
4.84.1 Compatibility considerations 472
4.84.2 Scene Actuator Configuration Set Command 472
4.84.3 Scene Actuator Configuration Get Command 473
4.84.4 Scene Actuator Configuration Report Command 474
4.85 Scene Controller Configuration Command Class, version 1 476
4.85.1 Compatibility Considerations 476
4.85.2 Scene Controller Configuration Set Command 476
4.85.3 Scene Controller Configuration Get Command 477
4.85.4 Scene Controller Configuration Report Command 478
4.86 Schedule Command Class, version 1 479
4.86.1 Terminology 479
4.86.2 Interoperability considerations 481
4.86.3 Schedule Supported Get Command 482
4.86.4 Schedule Supported Report Command 483
4.86.5 Schedule Set Command 488
4.86.6 Schedule Get Command 493
4.86.7 Schedule Report Command 494
4.86.8 Schedule Remove Command 495
4.86.9 Schedule State Set Command 496
4.86.10 Schedule State Get Command 497
4.86.11 Schedule State Report Command 497
4.87 Schedule Command Class, version 2 500
4.87.1 Compatibility considerations 500
4.87.2 Schedule Supported Get Command 501
4.87.3 Schedule Supported Report Command 502
4.87.4 Schedule Set Command 504
4.87.5 Schedule Get Command 505
4.87.6 Schedule Report Command 506
4.87.7 Schedule Remove Command 507
4.87.8 Schedule State Set Command 508
4.87.9 Schedule State Get Command 509
4.87.10 Schedule State Report Command 510
4.88 Schedule Command Class, version 3 510
4.88.1 Terminology 510
4.88.2 Compatibility considerations 510
4.88.3 Schedule Supported Report Command 511
4.88.4 Schedule Set Command 514

4.88.5 Schedule Get Command....517
4.88.6 Schedule Report Command....518
4.89 Schedule Command Class, version 4....520
4.89.1 Compatibility considerations....520
4.89.2 Schedule Supported Report Command....520
4.89.3 Schedule Supported Commands Get Command....522
4.89.4 Schedule Supported Commands Report Command....523
4.89.5 Schedule Set Command....525
4.89.6 Schedule Report Command....526
4.89.7 Schedule State Report Command....527
4.90 Schedule Entry Lock Command Class, version 1 [DEPRECATED]....529
4.90.1 Schedule Entry Lock Enable Set Command....529
4.90.2 Schedule Entry Lock Enable All Set Command....530
4.90.3 Schedule Entry Lock Supported Get Command....530
4.90.4 Schedule Entry Lock Supported Report Command....531
4.90.5 Schedule Entry Lock Week Day Schedule Set Command....532
4.90.6 Schedule Entry Lock Week Days Schedule Get Command....533
4.90.7 Schedule Entry Lock Week Day Schedule Report Command....534
4.90.8 Schedule Entry Lock Year Day Schedule Set Command....535
4.90.9 Schedule Entry Lock Year Day Schedule Get Command....537
4.90.10 Schedule Entry Lock Year Day Schedule Report Command....538
4.91 Schedule Entry Lock Command Class, version 2 [DEPRECATED]....538
4.91.1 Schedule Entry Lock Time Offset Get Command....539
4.91.2 Schedule Entry Lock Time Offset Set Command....539
4.91.3 Schedule Entry Lock Time Offset Report Command....540
4.92 Schedule Entry Lock Command Class, Version 3 [DEPRECATED]....540
4.92.1 Schedule Entry Type Supported Report Command....541
4.92.2 Schedule Entry Lock Daily Repeating Set Command....542
4.92.3 Schedule Entry Lock Daily Repeating Get Command....543
4.92.4 Schedule Entry Lock Daily Repeating Report....544
4.93 Screen Attributes Command Class, version 1....545
4.93.1 Screen Attributes Get Command....545
4.93.2 Screen Attributes Report Command....545
4.94 Screen Attributes Command Class, version 2....546
4.94.1 Screen Attributes Report Command....546
4.95 Screen Meta Data Command Class, version 1....547
4.95.1 Screen Meta Data Get Command....548
4.95.2 Screen Meta Data Report Command....549
4.96 Screen Meta Data Command Class, version 2....552
4.96.1 Screen Meta Data Report Command....552
4.97 Sensor Configuration Command Class, version 1 [OBSOLETED]....555
4.97.1 Sensor Trigger Level Set Command....556
4.97.2 Sensor Trigger Level Get Command....557
4.97.3 Sensor Trigger Level Report Command....557
4.97.4 Mapping example....558
4.98 Simple AV Control Command Class, version 1-4....558
4.98.1 Simple AV Control Set Command....558

4.98.2 Simple AV Control Get Command 560
4.98.3 Simple AV Control Report Command 560
4.98.4 Simple AV Control Supported Get Command 561
4.98.5 Simple AV Control Supported Report Command 561
4.99 Sound Switch Command Class, version 1 562
4.99.1 Terminology 562
4.99.2 Compatibility considerations 562
4.99.3 Sound Switch Tones Number Get Command 562
4.99.4 Sound Switch Tones Number Report Command 563
4.99.5 Sound Switch Tone Info Get Command 563
4.99.6 Sound Switch Tone Info Report Command 564
4.99.7 Sound Switch Configuration Set Command 565
4.99.8 Sound Switch Configuration Get Command 566
4.99.9 Sound Switch Configuration Report Command 566
4.99.10 Sound Switch Tone Play Set Command 567
4.99.11 Sound Switch Tone Play Get Command 568
4.99.12 Sound Switch Tone Play Report Command 568
4.100 Sound Switch Command Class, version 2 568
4.100.1 Compatibility Considerations 568
4.100.2 Sound Switch Tone Play Set Command 569
4.100.3 Sound Switch Tone Play Report Command 570
4.101 Tariff Table Configuration Command Class, version 1 571
4.101.1 Tariff Table Supplier Set Command 571
4.101.2 Tariff Table Set Command 574
4.101.3 Tariff Table Remove Command 575
4.102 Tariff Table Monitor Command Class, version 1 575
4.102.1 Tariff Table Supplier Get Command 575
4.102.2 Tariff Table Supplier Report Command 576
4.102.3 Tariff Table Get Command 577
4.102.4 Tariff Table Report Command 577
4.102.5 Tariff Table Cost Get Command 578
4.102.6 Tariff Table Cost Report Command 580
4.103 Thermostat Fan Mode Command Class, version 1 582
4.103.1 Thermostat Fan Mode Set Command 582
4.103.2 Thermostat Fan Mode Get Command 582
4.103.3 Thermostat Fan Mode Report Command 583
4.103.4 Thermostat Fan Mode Supported Get Command 583
4.103.5 Thermostat Fan Mode Supported Report Command 584
4.104 Thermostat Fan Mode Command Class, Version 2 584
4.104.1 Thermostat Fan Mode Set Command 584
4.104.2 Thermostat Fan Mode Report Command 585
4.105 Thermostat Fan Mode Command Class, Version 3 585
4.105.1 Thermostat Fan Mode Set Command 585
4.105.2 Thermostat Fan Mode Get Command 586
4.105.3 Thermostat Fan Mode Report Command 586
4.106 Thermostat Fan Mode Command Class, Version 4-5 586
4.106.1 Thermostat Fan Mode Set Command 587

4.106.2 Thermostat Fan Mode Get Command 589
4.106.3 Thermostat Fan Mode Report Command 589
4.106.4 Thermostat Fan Mode Supported Get Command 589
4.106.5 Thermostat Fan Mode Supported Report Command 590
4.107 Thermostat Fan State Command Class, version 1-2 590
4.107.1 Compatibility considerations 590
4.107.2 Thermostat Fan State Get Command 591
4.107.3 Thermostat Fan State Report Command 592
4.108 Thermostat Mode Command Class, version 1-2 592
4.108.1 Interoperability considerations 592
4.108.2 Thermostat Mode Set Command 593
4.108.3 Thermostat Mode Get Command 593
4.108.4 Thermostat Mode Report Command 593
4.108.5 Thermostat Mode Supported Get Command 594
4.108.6 Thermostat Mode Supported Report Command 594
4.109 Thermostat Mode Command Class, version 3 595
4.109.1 Compatibility considerations 595
4.109.2 Interoperability considerations 595
4.109.3 Thermostat Mode Set Command 596
4.109.4 Thermostat Mode Report Command 599
4.110 Thermostat Operating State Command Class, version 1 600
4.110.1 Thermostat Operating State Get Command 600
4.110.2 Thermostat Operating State Report Command 600
4.111 Thermostat Operating State Command Class, version 2 600
4.111.1 Thermostat Operating State Get 600
4.111.2 Thermostat Operating State Report 601
4.111.3 Thermostat Operating State Logging Supported Get 602
4.111.4 Thermostat Operating State Logging Supported Report 602
4.111.5 Thermostat Operating State Logging Get 603
4.111.6 Thermostat Operating State Logging Report 603
4.112 Thermostat Setback Command Class, version 1 605
4.112.1 Thermostat Setback Set Command 605
4.112.2 Thermostat Setback Get Command 606
4.112.3 Thermostat Setback Report Command 607
4.113 Thermostat Setpoint Command Class, version 1-2 607
4.113.1 Interoperability Considerations 607
4.113.2 Thermostat Setpoint Set Command 608
4.113.3 Thermostat Setpoint Get Command 609
4.113.4 Thermostat Setpoint Report Command 610
4.113.5 Thermostat Setpoint Supported Get Command 611
4.113.6 Thermostat Setpoint Supported Report Command 612
4.114 Thermostat Setpoint Command Class, version 3 612
4.114.1 Compatibility considerations 612
4.114.2 Interoperability Considerations 613
4.114.3 Thermostat Setpoint Set Command 614
4.114.4 Thermostat Setpoint Get Command 616
4.114.5 Thermostat Setpoint Report Command 617

4.114.6 Thermostat Setpoint Supported Get Command 618
4.114.7 Thermostat Setpoint Supported Report Command 619
4.114.8 Thermostat Setpoint Capabilities Get Command 620
4.114.9 Thermostat Setpoint Capabilities Report Command 621
4.115 User Code Command Class, version 1 622
4.115.1 Interoperability considerations 622
4.115.2 User Code Set Command 623
4.115.3 User Code Get Command 624
4.115.4 User Code Report Command 625
4.115.5 Users Number Get Command 625
4.115.6 Users Number Report Command 625
4.116 User Code Command Class, version 2 626
4.116.1 Terminology 626
4.116.2 Compatibility considerations 627
4.116.3 Interoperability considerations 628
4.116.4 Security considerations 628
4.116.5 User Code Set Command 629
4.116.6 Users Number Report Command 630
4.116.7 User Code Capabilities Get Command 630
4.116.8 User Code Capabilities Report Command 631
4.116.9 User Code Keypad Mode Set Command 634
4.116.10 User Code Keypad Mode Get Command 636
4.116.11 User Code Keypad Mode Report Command 636
4.116.12 Extended User Code Set Command 637
4.116.13 Extended User Code Get Command 642
4.116.14 Extended User Code Report Command 644
4.116.15 Master Code Set Command 646
4.116.16 Master Code Get Command 647
4.116.17 Master Code Report Command 648
4.116.18 User Code Checksum Get Command 649
4.116.19 User Code Checksum Report Command 650
4.117 Window Covering Command Class, version 1 650
4.117.1 Terminology 651
4.117.2 Compatibility considerations 652
4.117.3 Window Covering Parameters 653
4.117.4 Window Covering Supported Get Command 657
4.117.5 Window Covering Supported Report Command 657
4.117.6 Window Covering Get Command 658
4.117.7 Window Covering Report Command 658
4.117.8 Window Covering Set Command 659
4.117.9 Window Covering Start Level Change Command 660
4.117.10 Window Covering Stop Level Change Command 661
REFERENCES 664

1 Abbreviations

Abbreviation	Explanation
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	Domain Name System
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
SUC	Static Update Controller
TZO	Time Zone Offset
Unicode	Unicode is a standard for encoding of characters. (http://www.unicode.org/)
UTC	Coordinated Universal Time (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

Commands classes are divided in four categories:

- Application Command Classes
- Management Command Classes [15]
- Transport-Encapsulation Command Classes [14]
- Network-Protocol Command Classes [16]

The list of defined Command Classes with their associated category is available in [13].

This document describes the Command Classes designed for Application specific purposes.

2.1 Precedence of definitions

Device Class, Device Type and Command Class Specifications approved as final version during the Device Class, Device Type and Command Class Open Review process have precedence over this document until integrated into this document.

2.2 Terms used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document MUST be interpreted as described in IETF RFC 2119 [6].

Statements containing the IETF RFC 2119 [6] key words are at times marked with unique requirement numbers in the margin. The requirements numbers have the following syntax: CC:xxxx.xx.xx.xx.xxx with each x being an hexadecimal digit.

This document defines functionality as deprecated or obsoleted.

The term "obsolete" means that the functionality MUST NOT be supported in new implementations applying for certification.

A controller SHOULD provide controlling capabilities of the actual functionality for backwards compatibility with legacy devices.

The term "deprecated" also indicates an obsolete definition, but it permits new implementations applying for certification.

Thus, the term “deprecated” means that the functionality SHOULD NOT be supported in new implementations applying for certification. Often, another substitute functionality is REQUIRED if the deprecated functionality is implemented.

A controller SHOULD provide controlling capabilities of the actual functionality for backwards compatibility with legacy devices.

3 Command Class Overview

The following sections present a general Command Class overview and their associated rules. The requirements and recommendations apply to all Command Classes.

3.1 Overview

The Device Class specification [1] and Z-Wave Plus Device Type [10], [22] specification indicate combinations of command classes which MUST be supported for a particular product class or type.

CC:0000.00.00.11.001

3.2 Command class format

3.2.1 Frame format

All commands classes have a common header consisting of a Command Class identifier and a Command identifier. Each command can have from zero to N bytes of command data. The bit numbering starts at zero for the least significant bit. LSB is denoted as 'Bit 0' and MSB is denoted as 'Bit 7 throughout the document.

LSB and MSB notations are also used for multi bytes fields, indicating which byte is the most significant.

Figure 1, Generic command format and Figure 2, Generic extended command format 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

3.2.1.1 Command class

The Command Class identifier range is shown in Table 1, Command Class identifier range.

Table 1, Command Class identifier range

CC:0000.00.00.13.001

Command Class	Description
0x00	No Operation. Used by Z-Wave Protocol. MAY be used by the application.
0x01..0x1F	Reserved for the Z-Wave protocol
0x20..0xEE	Application Command Classes
0xEF	Support/Control Mark
0xF0	Non interoperable
0xF100	Security Mark
0xF101..0xFFFF	Extended Application Command Classes (2 byte-long Command Class identifier)

A Command Class can contain up to 255 different Commands. If the Command Class field is in the range 0xF1..0xFF, the Command Class identifier is therefore 2 bytes long. This allows for future extensions of the Command Classes and provides the possibility of having more than 4000 Command Classes.

3.2.1.2 Command

The command field contains the specific command indicating a node how to parse and interpret the command data. The command field length is always 1 byte.

3.2.1.3 Command data (N bytes)

The command data field contains data related to the command. The description for the command data is defined in each individual command.

Simple commands, such as Get commands, usually contain no command data. Other commands, such as Set or Report commands can contain several bytes of command data.

3.2.2 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:

CC:0000.00.00.11.002

- A node MUST NOT discard a frame based on the length field. A receiving node MUST use the 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.

CC:0000.00.00.11.003

- All implementations of a Command Class version higher than 1 MUST initialize all parameters associated with the version higher than 1. Thereby a node implementing the version 2 of a Command Class can interpret a version 1 received Command.

CC:0000.00.00.11.004

- A device supporting a Command Class having a version higher than 1 MUST support the Version Command Class [15] to be able to identify the supported version. If a node does not support the Version Command Class at its highest security level, it can be assumed that all Command Classes version are equal to 1.
- It is allowed to certify nodes supporting an older version of a Command Class despite a newer version exists as long as the generic/specific device specification does not require a specific version implemented.

3.3 Controlled and Supported Command Classes

A node can support and/or control a given Command Class.

If a Command Class is supported:
The node implements all the Command Class functionalities and can be set and read back by other nodes. When a Command Class is supported, it is REQUIRED to implement the whole Command Class.

CC:0000.00.00.11.005

If a Command Class is controlled:
The node implements the ability to interview, read and/or set other nodes supporting the Command Class. Nodes controlling Command Classes may use only a subset of the Commands within a Command Class (for example only Set commands). Even if using a Command Class partially for control, the use must comply with the Command Class requirements.

For example, a Set Command sent to Association Group destinations is a form of Command Class control.

3.4 Node Information Frame

The Node Information Frame (NIF) is used to inform other devices about the node capabilities. The NIF contains a structure with a protocol specific part that is handled by the Z-Wave protocol and an application specific part that is filled in by the application. The protocol specific part consists of a bit telling if the node is a continuously listening device, the Basic Device Class the node is based on etc. The application specific part consists of the Generic and Specific Device Class and the Command Classes that are supported and/or controlled by the node.

Generic and Specific Device Class are defined in [1] for legacy Z-Wave nodes or in [10], [22] for Z-Wave Plus nodes.

A NIF will be sent to the controller when a node is to be included in the network, excluded from the network or upon request.

Table 2, Node Information Frame Format shows the NIF structure.

Table 2, Node Information Frame Format

Byte descriptor \ bit number	7	6	5	4	3	2	1	0
Capability	Liste-ning	Routing	Max Speed			Protocol version		
Security	Opt. Func.	Sensor 1000ms	Sensor 250ms	Beam capabi-lity	Routing Slave	Specific Device	Con-troller	Secu-rity
Reserved	Z-Wave Protocol Specific Part							
Basic *)	Basic Device Class (Z-Wave Protocol Specific Part)							
Generic	Generic Device Class							
Specific	Specific Device Class							
NodeInfo[0]	Command Class 1							
…	…							
NodeInfo[n-1]	Command Class N							

*) The “Basic” field is only included when the NIF is sent by a controller

The Z-Wave Protocol in a controller saves all the Node Information except the supported and controlled
CC:0000.00.00.11.006 Command Classes when a node is included in the network. The reserved field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

3.4.1 Z-Wave Protocol Specific Part

The protocol specific part of the NIF is handled by the Z-Wave protocol. This information is automatically inserted in the packet by the protocol layer when transferring data using the API.

Basic Device Class

The Basic Device Class field contains an identifier that identifies what Basic Device Class this node is based on and is set by the Z-Wave protocol. A detailed description of all available Basic Device Classes is given in [1] for Z-Wave devices and [10], [22] for Z-Wave Plus devices. The Z-Wave Plus devices have an additional parameter Role Type defining device role in the network. The Role Type parameter is announced via the Z-Wave Plus Info Command Class [15].

CC:0000.00.00.11.007 This field is only included when the NIF frame is sent by a controller. A slave MUST omit this field. Refer to [9] for controller and slave definition.

3.4.2 Application Specific Part

The application specific part of the NIF is handled by the application. The information must be in accordance with the defined classes to obtain interoperability.

Listening Flag

The Listening flag is used to indicate that the node is always listening if set to 1. An always listening node must be powered continuously and reside on a fixed position in the installation. An always listening node is included in the routing table to assist as repeater in the network. The routing table is static during normal operation. In case the Listening flag is set to 0, the node is non-listening. This is typically used for battery operated nodes being asleep when the protocol is idle to prolong battery lifetime. A battery operated node is not included in the routing table and is not used as a router in the network. In some instances the node's position in the network is still determined, and stored by the protocol.

Optional Functionality Flag

The Optional Functionality flag is used to indicate that this node supports other command classes than the mandatory for the selected generic/specific device class and that a controlling node needs to look at the supported command classes to fully control this device.

Generic Device Class

The Generic Device Class field contains an identifier that identifies what Generic Device Class this node is part of and must be set by the application. For a detailed description of all available Generic Device Classes, refer to [1] for Z-Wave devices and [10], [22] for Z-Wave Plus devices.

Specific Device Class

The Specific Device Class field specifies what Specific Device Class this application is part of and must be set by the application. For a detailed description of all available Specific Device Classes, refer to [1] for Z-Wave devices and [10], [22] for Z-Wave Plus devices.

Command Class

The Command Class field is used to advertise Command Classes implemented by the node. The field
CC:0000.00.00.11.008 MUST NOT be longer than 35 bytes.

CC:0000.00.00.11.009 The field MUST advertise the list of Command Classes that the node supports.

CC:0000.00.00.13.002 The field MAY advertise the list of Command Classes that the node can control in other nodes. If
CC:0000.00.00.11.00A present, the list of controlling Command Classes MUST be prepended by the COMMAND_CLASS_MARK
CC:0000.00.00.12.004 Command Class identifier. It is NOT RECOMMENDED to advertise controlled Command Classes.

It has been found that legacy controllers may read as little as 6 lines from this list. For backwards
CC:0000.00.00.12.001 compatibility, the list SHOULD advertise supported command classes in the order indicated in Table 3, NIF :: Command Class advertising priorities :

Table 3, NIF :: Command Class advertising priorities

Priority	Command Class
1 (First line)	COMMAND_CLASS_ZWAVEPLUS_INFO Applies only to Z-Wave Plus products
2 (if supported)	COMMAND_CLASS_SWITCH_MULTILEVEL or COMMAND_CLASS_SWITCH_BINARY
3 (if supported)	COMMAND_CLASS_SWITCH_ALL
4 (if supported)	COMMAND_CLASS_ASSOCIATION
5	All other command classes

Table 4, NIF::Command Class list structure shows the Command Class list structure of the Node Information frame:

Table 4, NIF::Command Class list structure

Description	Command Class list field content							
	7	6	5	4	3	2	1	0
Non-secure Supported Command Classes	Command Class 1 *)							
	…							
	Command Class M *)							
Support/Control Mark	0xEF							
Non-secure Controlled Command Classes	Command Class 1 *)							
	…							
	Command Class K *)							

*) Command classes may be extended ⇒ spanning two bytes for one command class

CC:0000.00.00.13.003 The Command Class list MAY start or finish with the identifier COMMAND_CLASS_MARK.

3.4.3 NIF and Multi Channel/Security Command Classes

CC:0000.00.00.11.00B The NIF MUST represent command classes supported without Multi Channel or Security encapsulation [14].

After inclusion in a Z-Wave network, the NIF represents the node's Command Class capabilities when using no Security and no Multi Channel encapsulation.

Multi Channel Root Devices advertise their capabilities via the NIF, but they advertise the Command Class capabilities of their End Points via the Multi Channel Command Class.

Security bootstrapped nodes advertise their capabilities using security encapsulation via the Security 0 or Security 2 Command Class.

The Command Class list advertised in the NIF may vary depending on the inclusion state and S0/S2 bootstrapping state of the node.

CC:0000.00.00.11.00C A node supporting security (S0 and/or S2) MUST advertise command classes in the NIF according to Table 5, NIF content depending on inclusion and security bootstrapping.

Table 5, NIF content depending on inclusion and security bootstrapping

Network inclusion	Security bootstrapping	Command Classes advertised in the NIF
Not included in a network	N/A	At least all Command Classes that MUST always be non-securely supported. All supported Command Classes MAY be advertised
Included in a network	No security supported by the included node	All supported Command Classes
Included in a network	Before the node times out waiting for Security bootstrapping	At least all Command Classes that MUST always be non-securely supported. All supported Command Classes MAY be advertised
Included in a network	Timed out waiting for Security Bootstrapping (i.e. the process never started)	The same Command Class list as when security bootstrapping is successful but no key was granted[1]
Included in a network	Security bootstrapping failed. (i.e. the process started but did not complete without error)	All Command Classes that MUST always be non-securely supported Command Classes that are supported when no key is granted MAY be added or removed from the list.[1]
Included in a network	Bootstrapped with S0/S2	All Command Classes that MUST always be non-securely supported only. (i.e. Command Classes supported even if not using S0/S2 encapsulation).

[1] Refer to each individual Device Type [10], [22] for the list of Command Classes supported when no security key is granted

CC:0000.00.00.11.00D Command Classes advertised as supported in the NIF after S0/S2 bootstrapping MUST also be supported at higher security levels unless encapsulated outside security.

3.4.3.1 Examples

In the case of a node requesting Access Control, they require a certain security level before supporting their command class. For example a Door Lock Device Type will support its command classes as shown in Figure 3

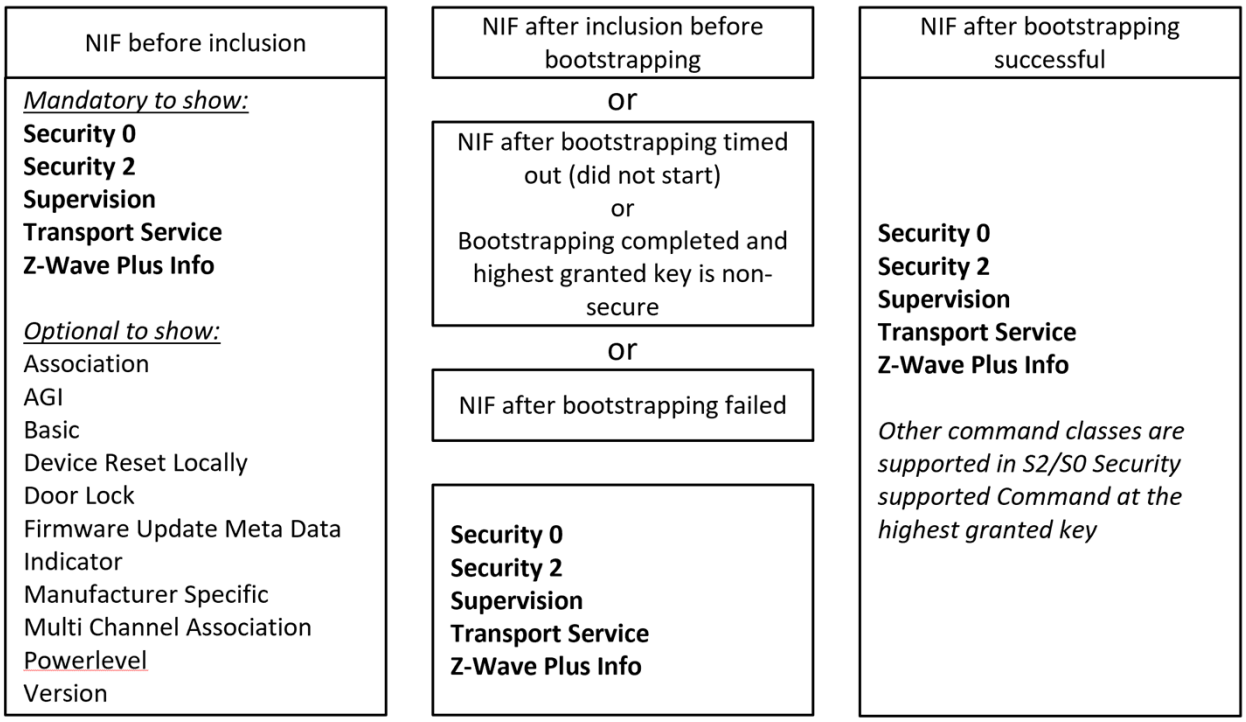


Figure 3, Access Control node NIF contents (always non-secure Command Classes shown in bold)

Nodes requesting a lower security level than Access Control have more freedom about advertising their regular command classes in the different scenarios. An example of a Binary Switch Device Type is shown in Figure 4

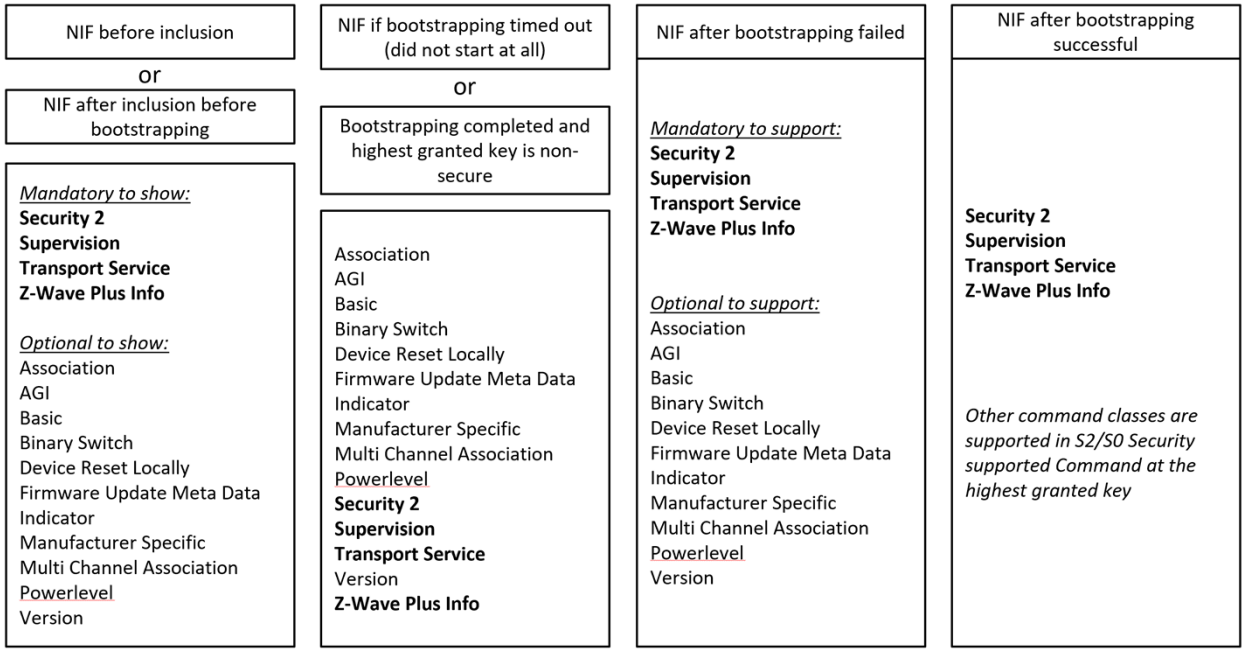


Figure 4, Non-Access Control node NIF contents (always non-secure Command Classes shown in bold)

3.4.4 Command Class specific NIF rules

CC:0000.00.00.13.004 A given Command Class MAY have additional requirement with respects to the NIF, Multi Channel Capability Report or S0/S2 Commands Supported Report.

For example, the Basic Command Class is never advertised in the NIF or the Z-Wave Plus Info is always in the NIF if supported.

In this case, requirements are listed in each individual Command Class definition, in the compatibility considerations section.

3.5 Multicast and broadcast commands

CC:0000.00.00.13.006 A node MAY send a command to several destinations using a Multicast or Broadcast frame. This is allowed only if the actual command does not require the responding node to return a response.

Unless specified otherwise for a particular command:

CC:0000.00.00.11.00E
- Commands requiring another command to be returned in response by a receiving node MUST NOT be issued via multicast addressing in a Z-Wave network

CC:0000.00.00.11.00F
- A receiving node MUST NOT return a response if a command is received via multicast addressing.

The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

3.6 Actuator Control

An actuator device may support one or more actuator command classes. This section presents terminology and conventions applying to all actuator command classes.

The actuator control group comprises the following command classes:

- Barrier Operator Command Class
- Basic Command Class
- Binary Switch Command Class
- Color Switch Command Class
- Door Lock Command Class
- Multilevel Switch Command Class
- Simple AV Command Class
- Sound Switch Command Class
- Thermostat Setpoint Command Class
- Thermostat Mode Command Class
- Window Covering Command Class

3.6.1 Terminology

Actuators may be controlled in two possible manners.

A position actuator responds to a position control command, which specifies a target value and optionally a duration for the transition from the current value to the target value. The value may be binary or multilevel.
State control commands may be encoded as special multilevel position control values, e.g. 0xFF for “On” and 0x00 for “Off”. The most recent (non-zero) target value is restored in response to the “On” state control command.

A motion actuator responds to start/stop commands. A motion control command specifies a direction, optionally a start value, and a transition rate.

3.6.2 Reporting values

A position control command may be used to initiate a transition to a new target value. A position report advertises the current value of the device hardware, optionally the target value and the remaining transition duration. If the transition is initiated by a motion control command, the reported target value is the min or max value and the duration is the time needed to reach the target value at the actual transition rate.

A controlling device should not assume that the position value advertised in a Report is identical to a value previously issued with a position control command when a transition has ended.

CC:0000.00.00.53.002 A controlling node MAY want to receive application-level acknowledgements after issuing actuating commands. It is the responsibility and choice of the controlling node whether or not to ensure that a
CC:0000.00.00.52.001 actuating command has been successfully executed. If doing so, it is RECOMMENDED to use the Supervision Command Class [14] or MAY read back the node’s state at a later time with a Get Command.

CC:0000.00.00.12.005 A supporting node receiving an actuating Command SHOULD NOT return a Report Command advertising its status or level unless specified in the Command Class specification.

3.6.3 Command values vs. hardware values

A device may implement fewer hardware levels than supported by a position control Command Class. The hardware levels should be distributed uniformly over the entire range. The mapping of command
CC:0000.00.00.11.010 values to hardware levels must be monotonous, i.e. a higher value MUST be mapped to either the same or a higher hardware level. An example is found in Table 6, Mapping command values to hardware levels (example).

Table 6, Mapping command values to hardware levels (example)

Multilevel Value	Hardware level	State
0	0	Off
1..33	33%	On
34..66	66%	On
67..99	100%	On

3.6.4 Supporting multiple actuator Command Classes

CC:0000.00.00.13.007 Several actuator Command Classes MAY co-exist in a node or Multi Channel End Point. For example, a
CC:0000.00.00.13.008 node MAY support both Window Covering Command Class and Multilevel Switch Command Class.

CC:0000.00.00.12.002 In this case, the two actuator Command Classes SHOULD actuate the same resource. If a node
CC:0000.00.00.12.003 implements several actuating resources, e.g. a LED and a power switch, it is RECOMMENDED to use Multi Channel End Points for each individual resource.

3.7 Common fields and encoding

3.7.1 Reserved and Res fields

CC:0000.00.00.11.011 Fields named ‘Reserved’ or ‘Res’ MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

3.7.2 Reserved values and reserved bits

CC:0000.00.00.11.012 Values of fields in commands that are marked as “reserved” or “res” MUST NOT be used by devices sending commands and MUST be ignored by devices receiving commands.

CC:0000.00.00.11.013 Bits in commands that are marked as “reserved” or “res” MUST be set to 0 by devices sending commands and MUST be ignored by devices receiving commands.

CC:0000.00.00.11.014 Undefined values for a given parameter MUST be treated as “reserved”. Reserved values MUST NOT be used by a sending node and MUST be ignored by a receiving node.

3.7.3 Duration encoding

CC:0000.00.00.11.015 Some actuator command classes allow controlling nodes to specify a duration for reaching the target value. The duration MUST be encoded according to Table 7, Actuator Command Class Duration Set encoding.

Table 7, Actuator Command Class Duration Set encoding

Value	Description
0x00	Instantly
0x01..0x7F	1 second (0x01) to 127 seconds (0x7F) in 1 second resolution.
0x80..0xFE	1 minute (0x80) to 127 minutes (0xFE) in 1 minute resolution.
0xFF	Factory default duration.

CC:0000.00.00.11.016 A node supporting an actuator Command Class may advertise the duration left to reach the target value advertised in a report. The duration MUST be encoded according to Table 8, Actuator Command Class Duration Report encoding.

Table 8, Actuator Command Class Duration Report encoding

Value	Description
0x00	0 seconds. Already at the Target Value.
0x01..0x7F	1 second (0x01) to 127 seconds (0x7F) in 1 second resolution.
0x80..0xFD	1 minute (0x80) to 126 minutes (0xFD) in 1 minute resolution.
0xFE	Unknown duration
0xFF	*Reserved*

3.7.4 Unsigned encoding

Unless specified otherwise in a field description, the field encoding is using unsigned representation.

CC:0000.00.00.11.017 Fields using unsigned encoding MUST comply with Table 9, Default unsigned encoding examples

Table 9, Default unsigned encoding examples

Value (hex)	Unsigned 8 bit representation (decimal)
0x00	0
0x01	1
0x02	2
0x7F	127
0x80	128
0xFE	254
0xFF	255

Value (hex)	Unsigned 16 bit representation (decimal)
0x0000	0
0x0001	1
0x0002	2
0x7FFF	32767
0x8000	32768
0xFFFE	65534
0xFFFF	65535

Value (hex)	Unsigned 32 bit representation (decimal)
0x00000000	0
0x00000001	1
0x00000002	2
0x7FFFFFFF	2147483647
0x80000000	2147483648
0xFFFFFFFE	4294967294
0xFFFFFFFF	4294967295

3.7.5 Signed encoding

CC:0000.00.00.11.018 Some fields use the signed encoding representation.Signed encoding is using the two's complement representation. Fields using signed encoding MUST comply with Table 10, Signed field encoding (two's complement representation).

Table 10, Signed field encoding (two's complement representation)

Value (hex)	Signed 8 bit representation (decimal)
0x7F	127
0x02	2
0x01	1
0x00	0
0xFF	-1
0xFE	-2
0x80	-128

Value (hex)	Signed 16 bit representation (decimal)
0x7FFF	32767
0x0002	2
0x0001	1
0x0000	0
0xFFFF	-1
0xFFFE	-2
0x8000	-32768

Value (hex)	Signed 32 bit representation (decimal)
0x7FFFFFFF	2147483647
0x00000002	2
0x00000001	1
0x00000000	0
0xFFFFFFFF	-1
0xFFFFFFFE	-2
0x80000000	-2147483648

3.7.6 Fields values and version

New Command Class versions may define new possible values for a field introduced previously. In that

CC:0000.00.00.11.019 case, a version column is added to the value table. A supporting node MUST support the values

associated its version and previous versions and MUST NOT support values associated to future versions. An example is given in Table 11, Field values and Command Class version (example).

Table 11, Field values and Command Class version (example)

Value	Description	Version
0x00	Value introduced in version 1, valid and mandatory in all versions	1
0x01	Value introduced in version 2, valid and mandatory in version 2 and newer Version 1 MUST consider this value as reserved	2
0x02	Value introduced in version 2, valid and mandatory in version 2 and newer Version 1 MUST consider this value as reserved	2
0xFF	Value introduced in version 3, valid and mandatory in version 3 and newer Version 1 and 2 MUST consider this value as reserved	3

4 Command Class Definitions

The following subchapters contain definitions of Application Command Classes.

4.1 Alarm Command Class, version 1 [DEPRECATED]

THIS COMMAND CLASS VERSION HAS BEEN DEPRECATED

CC:0071.01.00.12.001 A device MAY implement this Command Class version, but it is RECOMMENDED that new implementations comply with Notification Command Class, version 8

The Alarm Command Class allows applications to report alarm or service conditions. Since these

CC:0071.01.00.11.001 parameters are not standardized across devices the alarms/service parameters MUST be described in the user manual (or an installer manual).

4.1.1 Interoperability considerations

The Alarm Command Class has been superseded by the Notification Command Class. Refer to most recent version of Notification Command Class.

4.1.2 Alarm Get Command

This command is used to get the value of an alarm.

CC:0071.01.04.11.001 The Alarm Report Command MUST be returned in response to this command if the alarm type is supported.

CC:0071.01.04.11.002 This command MUST NOT be issued via multicast addressing.

CC:0071.01.04.11.003 A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

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.

4.1.3 Alarm Report Command

This command is used to report the type and level of an alarm.

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.

4.2 Alarm Command Class, version 2 [DEPRECATED]

THIS COMMAND CLASS VERSION HAS BEEN DEPRECATED

CC:0071.02.00.12.001 A device MAY implement this Command Class version, but it is RECOMMENDED that new implementations comply with Notification Command Class, version 8

The Alarm Command Class is 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 described in this version remain unchanged from Alarm Command Class, version 1.

4.2.1 Interoperability considerations

The interoperability considerations from version 1 also apply for this version. Refer to 4.1.1.

4.2.2 Alarm Set Command

This command is used to set the activity of the Z-Wave Alarm Type and Status.

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 bits)

Refer to 4.1.3 Alarm Report Command

Z-Wave Alarm Status (8 bits)

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.

CC:0071.02.06.11.001
CC:0071.02.06.11.002
CC:0071.02.06.12.001 **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 SHOULD be enabled.

CC:0071.02.06.11.003 Products that do not allow deactivation of a specific Alarm Type, MUST respond to a Alarm Configuration Set deactivating the Alarm Type in question by returning an Application Rejected Request Command of the Application Status Command Class.

4.2.3 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.

CC:0071.02.04.11.001 The Alarm Report Command MUST be returned in response to this command if the alarm type is supported.

CC:0071.02.04.11.002 This command MUST NOT be issued via multicast addressing.

CC:0071.02.04.11.003 A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

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 bits)

CC:0071.02.04.11.004 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 bits)

CC:0071.02.04.11.005 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.

CC:0071.02.04.11.006 CC:0071.02.04.12.001 A node receiving a non-supported Z-Wave Alarm Type MUST ignore the command. A controlling node SHOULD interview the device for supported Alarm Types by means of Alarm Type Supported Get Command prior to Alarm Get.

4.2.4 Alarm Report Command

This 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 bits)

Refer to 4.1.3 Alarm Report Command

Alarm Level (8 bits)

Refer to 4.1.3 Alarm Report Command

Zensor Net Source Node ID (8 bits)

CC:0071.02.05.11.001 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 bits)

Refer to 4.2.2 Alarm Set Command

Number of Event Parameters (8 bits)

Indicates the Number of Event Parameters fields used in bytes.

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

CC:0071.02.05.11.002 [18] specifies the Alarm Types and its subordinate parameters defined by the Z-Wave Alliance. The fields that do not contain any definition of the Z-Wave Alarm Type MUST be set to '0' in the Alarm Report Command.

CC:0071.02.05.11.003 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.

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

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.

4.2.5 Alarm Type Supported Get Command

This command is used to request the supported alarm types.

CC:0071.02.07.11.001 The Alarm Type Supported Report Command MUST be returned in response to an Alarm Type Supported Get command.

CC:0071.02.07.11.002 This command MUST NOT be issued via multicast addressing.

CC:0071.02.07.11.003 A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

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

4.2.6 Alarm Type Supported Report Command

This command is used to advertise the supported alarm types in the application.

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)

0 = the device implements only Notification CC V2 (or newer) Notification Type(s).
1 = the device implements Notification CC V2 Notification Types as well as proprietary Alarm CC V1 Alarm Types and Alarm Levels.

Reserved

CC:0071.02.08.11.001 This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Number of Bit Masks (5 bits)

Indicates the Number of Bit Masks fields used in bytes.

Bit Mask (N Bytes)

The Bit Mask fields describe the supported Z-Wave Alarm Types by the device.

CC:0071.02.08.11.002

- Bit 0 in Bit Mask 1 is not allocated to any Z-Wave Alarm Type and MUST therefore be set to zero.
- Bit 1 in Bit Mask 1 indicates if Z-Wave Alarm Type = 1 (Smoke) is supported.
- Bit 2 in Bit Mask 1 indicates if Z-Wave Alarm Type = 2 (CO) is supported.
- Bit 3 in Bit Mask 1 indicates if Z-Wave Alarm Type = 3 (CO_2) is supported
- …

CC:0071.02.08.11.003 If the Z-Wave Alarm Type is supported the corresponding bit MUST be set to 1. If the Z-Wave Alarm Type is not supported the corresponding bit MUST be set to 0.

CC:0071.02.08.11.004 Z-Wave Alarm Type = 0xFF (Return first Alarm on supported list) MUST NOT be advertised in the Bit Masks.

CC:0071.02.08.11.005 The number of Bit Mask fields MUST match the value advertised in the Number of Bit Masks field.

Note that the mapping of bit 1 to Alarm Type =1 differs from the support mapping used by the Multilevel Sensor Command Class. The Multilevel Sensor Command Class maps bit 0 to Sensor Type = 1.

4.3 Alarm Sensor Command Class, version 1 [DEPRECATED]

THIS COMMAND CLASS HAS BEEN DEPRECATED

A device MAY implement this command class, but it is RECOMMENDED that new implementations use the Notification Command Class.
If implementing this Command Class, it is RECOMMENDED that the Notification Command Class is also implemented.

The Alarm Sensor Command Class is used to realize Sensor Alarms.

4.3.1 Alarm Sensor Get Command

This command is used to request the status of a sensor.

The Alarm Sensor Report Command MUST be returned in response to this command if the sensor type is supported.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

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 Table 12. 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.

Table 12, Alarm Sensor Get::Sensor Type encoding

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

4.3.2 Alarm Sensor Report Command

This command is used to advertise the alarm state.

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)

Refer to 4.3.1 Alarm Sensor Get Command.
The Sensor Type 0xFF MUST NOT be advertised in this command.

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 0x65…0xFE are reserved and MUST be ignored by receiving nodes.

Seconds (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.

4.3.3 Alarm Sensor Supported Get Command

This command is used to request the supported sensor types from the device.

The Alarm Sensor Supported Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

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

4.3.4 Alarm Sensor Supported Report Command

This command is used to report 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_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 (N Bytes)

The Bit Mask fields describe the supported sensor types by the device.

- Bit 0 in Bit Mask 1 indicates if Sensor Type = 0 (General Alarm) is supported.
- Bit 1 in Bit Mask 1 indicates if Sensor Type = 1 (Smoke Alarm) is supported.
- …

The sensor type is supported if the bit is 1 and the opposite if 0. 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 MUST be determined from the length field in the frame.

Note that the mapping of bit 1 to Sensor Type =1 differs from the support mapping used by the Multilevel Sensor Command Class. The Multilevel Sensor Command Class maps bit 0 to Sensor Type = 1.

4.4 Alarm Silence Command Class, version 1

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

4.4.1 Alarm Silence Set Command

This command is 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)

Mode specifies the different options to silence sensor alarms. Modes are defined by the Z-Wave Alliance.

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

Seconds (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 (N Bytes)

The Bit Mask fields describe the sensor types to disable sounding from.

- Bit 0 in Bit Mask 1 indicates if Sensor Type = 0 (General Alarm) is disabled.
- Bit 1 in Bit Mask 1 indicates if Sensor Type = 1 (Smoke Alarm) is disabled.
- …

If the sensor type is disabled the bit MUST be set to 1. If the sensor type is not disabled the bit MUST be set to 0. 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.

4.5 All Switch Command Class, version 1 [OBSOLETED]

THIS COMMAND CLASS HAS BEEN OBSOLETED
New implementations MUST NOT support this Command Class. Controlling nodes MUST use S2 Multicast mechanism in order to achieve the All On and All Off functionality

The All Switch Command Class is used to switch all devices on or off. Devices may 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.

4.5.1 All Switch Set Command

This command is used to instruct 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_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.
0x03..0xFE	*Reserved*
0xFF	Indicates that the switch is included in the all on/all off functionality.

4.5.2 All Switch Get Command

This command is used to ask a device if it is included or excluded from the all on/all off functionality.

The All Switch Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

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

4.5.3 All Switch Report Command

This command is used to report if the device 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_REPORT							
Mode							

Mode (8 bits)

Refer to the All Switch Set Command.

4.5.4 All Switch On Command

This command is used to inform a switch that it SHOULD be turned on. A receiving device MUST NOT react to this command if the actual operation has been prohibited via the Switch All Set command. Like the Basic Set On command, this command MUST cause the device to restore the most recent (non-zero) level.

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

4.5.5 All Switch Off Command

This command is used to inform a switch that it SHOULD be turned off. A receiving device MUST NOT react to this command if the actual operation has been prohibited via the Switch All Set command.

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

4.6 Anti-theft Command Class, version 1 [OBSOLETED]

THIS COMMAND CLASS VERSION HAS BEEN OBSOLETED

New implementations MUST use the Anti-theft Command Class Version 2 or newer.

The Anti-theft Command Class is used to disable a subset of supported/controlled command classes in a device if the device is being excluded and re-included into a Z-Wave network again. This command class is typically used when installing a Z-Wave device in a public location such as a hotel room or conference center. The command class allows the user to lock the device to the actual Z-Wave network and to render it useless if it is removed from the local network without being unlocked. Another application would be to protect service provider owned products from leaving the service providers network before they are paid for.

Version 2 limits the Magic Code and Anti-theft Hint maximum bytes to 10. This makes it possible to embed Anti-theft Command Class Version 2 in one Security Command Class and thereby avoid splitting it.

4.7 Anti-theft Command Class, version 2 [DEPRECATED]

THIS COMMAND CLASS VERSION HAS BEEN DEPRECATED

New implementations SHOULD use the Anti-theft Command Class, version 3.

The Anti-theft Command Class MUST NOT be supported unless the Device Class or Device Type implemented by the device explicitly allows for support of the Anti-theft Command Class.

The Anti-theft Command Class is intended for devices operating in public locations such as hotel rooms or a conference center. The purpose of the Anti-theft Command Class is to render a device useless if it is removed from its actual network without being unlocked by the owner or a service provider.

The Anti-theft Command Class is used to disable all command classes related to the actual application functionality of a device if is it excluded and later included in another network. It does not matter if the device implements a single resource addressed via the Root Device or a collection of resources addressed via individual Multi Channel End Points Enabling anti-theft protection in a device MUST NOT change any operation with respect to supported/controlled command classes as long as the device stays in the actual network.

If a locked device is excluded, it MUST enter the protected state. When in the protected state, the node information frame (NIF) MUST NOT advertise support of the protected command classes.The NIF MUST however continue advertising support of the Anti-theft Command Class and all other non-application specific command classes; just as when the device operates in its home network.
The device MUST NOT respond to application commands while in the protected state. A device in protected state MUST NOT leave its protected state if it is re-included into its home network. Another Anti-Theft Set command MUST be used to either disable locking or to clear the protected state.

The following non-device specific command classes must not be protected by the anti-theft functionality (i.e. will always be available in the device regardless of protection state):

- Manufacturer Specific Command Class
- Version Command Class
- Anti-theft Command Class
- Security Command Class (optional)
  Security encapsulated command classes are allowed to be protected. In that case they must be removed from the *Security Commands Supported Report Command* when in the protected state in a foreign network.

The protection state may be updated by sending the set command with the correct magic code to the device at any time and in any network. When the protection state is updated the device must return to normal operation, regardless of whether the update is to disable or re-enable protection. However, it is not possible to update protection state when device is excluded because it must be able to receive a command.

The Anti-theft protection state must be preserved in the following situations:

- Exclusion of a network
- Reset to factory default
- OTA update of firmware

If secure device, supports Anti-theft Command Class, Security Command Class needs to be supported regardless of anti-theft protection state. A security enabled device MUST be able to join any secure Z-Wave network regardless of its anti-theft protection state.

It is RECOMMENDED for a device that supports anti-theft protection to have physical mark that indicates that this device is capable of being locked. It is further RECOMMENDED that the device is capable of signaling via a LED or other means if the device refuses inclusion in a network because the device is locked to another network. Finally, the user guide and installation manual MUST advertise support of anti-theft protection.

4.7.1 Anti-theft Set Command

This command is used to enable/disable anti-theft protection in a device already included into a Z-Wave Network by sending a magic code to device in question. The same magic code MUST be used to disable anti-theft protection again. A new magic code may be used the next time to enable anti-theft protection in the device, but only if protection is disabled at the time. A new device MUST have anti-theft protection disabled. Enabling anti-theft protection in an already-enabled device restores it to normal operation if it is in reduced functionality mode, but otherwise has no effect.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ANTITHEFT							
Command = ANTITHEFT_SET							
Enable	Number of Magic Code bytes						
Magic Code 1							
…							
Magic Code N							
Manufacturer ID MSB							
Manufacturer ID LSB							
Anti-theft Hint Number Bytes							
Anti-theft Hint Byte 1							
…							
Anti-theft Hint Byte N							

Enable (1 bit)

The value MAY be 0 (Attempt to disable anti-theft protection in device) or 1 (Attempt to enable or re-enable anti-theft protection in device). It is not necessary to first disable an exclude device having protection enable; it can be re-enabled directly in a new network by using correct magic code again.

Number of Magic Code bytes (7 bits)

Indicates the Number of Magic Code fields N used in bytes. Maximum number of Magic Code fields MUST NOT exceed 10 bytes.

Magic Code (N bytes)

The Magic Code fields hold the code to enable/disable the Z-Wave device in question.

Manufacturer ID (2 bytes)

The Manufacturer ID of the company's product having a central role in the application requiring anti-theft protection enabled. Device should report 0xFFFF if anti-theft protection is disabled.

Manufacturer identifiers can be found in [12].

Number of Anti-theft Hint bytes (8 bits)

Indicates the Number of Anti-theft Hint fields N used in bytes. If length is 0 no Hint provided. Maximum number of Anti-theft Hint fields MUST NOT exceed 10 bytes.

Anti-theft Hint Byte (N bytes)

Anti-theft Hint Bytes that may be used as an identifier or key value for retrieving the Magic Code. The exact format and meaning of these Bytes is specific to the product or service that enabled anti-theft protection on the device, as identified by the Manufacturer ID above. If it is necessary to render the Hint Bytes for display, each byte should be interpreted as an unsigned integer value and represented in hexadecimal.

4.7.2 Anti-theft Get Command

This command is used to get an Anti-theft Report Command showing status of the Z-Wave device in question.

The Anti-theft Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ANTITHEFT							
Command = ANTITHEFT_GET							

4.7.3 Anti-theft Report Command

This command is used to report status of the Z-Wave device in question.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ANTITHEFT							
Command = ANTITHEFT_REPORT							
Anti-theft Protection Status							
Manufacturer ID MSB							
Manufacturer ID LSB							
Anti-theft Hint Number Bytes							
Anti-theft Hint Byte 1							
…							
Anti-theft Hint Byte N							

Anti-theft Protection Status (8 bits)

Anti-theft Protection Status specifies the actual status of Z-Wave device in question. Refer to the table below with respect to defined status values.

Value	Anti-theft Protection Status
0x00	Reserved.
0x01	Anti-Theft Protection is currently DISABLED, and the Z-Wave Device is fully functional.
0x02	Anti-Theft Protection is currently ENABLED, and the Z-Wave Device is fully functional.
0x03	Anti-Theft Protection is currently ENABLED, and the Z-Wave Device is NOT fully functional (i.e., the Device was excluded from a network without disabling protection, and an ANTITHEFT_SET command with the correct Magic Code has not yet been received in the current network).
0x04..0xFF	Reserved

Manufacturer ID (2 bytes)

The Manufacturer ID of the company's product having a central role in the application requiring anti-theft protection enabled. Device should report 0xFFFF if anti-theft protection is disabled.

Number of Anti-theft Hint bytes (8 bits)

Indicates the Number of Anti-theft Hint fields N used in bytes. If length is 0 no Hint provided. Maximum number of Anti-theft Hint fields MUST NOT exceed 10 bytes.

Anti-theft Hint Byte (N bytes)

Anti-theft Hint Bytes. See the Anti-Theft Set Command for more details.

4.7.4 Examples

Following examples are for reference only.

4.7.4.1 Example of a non-secure Thermostat

Below is shown an example of the Node Information Frame (NIF) content for a non-secure thermostat. The first NIF shows a device having anti-theft protection disabled. The device may never been included into a network or reside in a network or excluded from a network:

Disabled anti-theft protection
Manufacturer Specific Command Class
Version Command Class
Anti-theft Command Class
Thermostat Operating State Command Class
Thermostat Mode Command Class
Association Command Class
Battery Command Class

The second NIF shows a device having anti-theft protection enabled. The device may be excluded from network in which it was originally anti-theft protection enabled or re-included into a network. This also applies in case device is re-included into the network, which device originally was anti-theft protection enabled:

Enabled anti-theft protection
Manufacturer Specific Command Class
Version Command Class
Anti-theft Command Class

4.7.4.2 Example of a security enabled Thermostat

Below is shown an example of the Node Information Frame (NIF) content for a security enabled thermostat. The first NIF shows a device having anti-theft protection disabled. The device may never been included into a network or reside in a network or excluded from a network:

Disabled anti-theft protection
Manufacturer Specific Command Class
Version Command Class

Disabled anti-theft protection
Security Command Class

Finally, the *Security Commands Supported Report Command* reports support of the following command classes:

- Anti-theft Command Class
- Thermostat Operating State Command Class
- Thermostat Mode Command Class
- Association Command Class
- Battery Command Class

The second NIF shows a device having anti-theft protection enabled. The device may be excluded from network in which it was originally anti-theft protection enabled or re-included into a network. This also applies in case device is re-included into the network, which device originally was anti-theft protection enabled. The NIF is unchanged because all application oriented command classes are security encapsulated except the default command classes:

Enabled anti-theft protection
Manufacturer Specific Command Class
Version Command Class
Security Command Class

Finally, the *Security Commands Supported Report Command* reports support of at least the Anti-theft Command Class to be able to disable anti-theft protection:

- Anti-theft Command Class

The Anti-theft Command Class is supported securely making malicious attempts to enable anti-theft protection very difficult.

4.8 Anti-theft Command Class, version 3

This Command Class is used to lock (and possibly unlock) a node.

4.8.1 Compatibility Considerations

4.8.1.1 Command Class dependencies

A node supporting this Command Class MUST also support the Anti-Theft Unlock Command Class, version 1.

4.8.1.2 Lock/Unlock requirements

The Anti-Theft Command Class is intended for devices operating in public locations such as hotel rooms or a conference center. The purpose of the Anti-theft Command Class is to render a device useless if it is removed from its actual network without being unlocked by the owner or a service provider.

A node can be locked (or protected) or unlocked.

The unlocked state means that the node will operate normally in any Z-Wave network.

The locked state means that the node will restrict usage of most of its command classes after being excluded from a network, so that it cannot be used by any controlling unit. The locked state MUST persist after the following operations:

- Network inclusion/exclusion
- Factory reset to default.
- Firmware upgrade

When a node is locked and either excluded from a network or factory reset to default, it will enter its restricted mode. In the restricted mode, the node will not grant access to its application functionalities. The restricted mode MUST stop when the node is unlocked again.

When a node runs in the restricted mode, the NIF MUST adapt its content to reflect the currently supported Command Classes. The following Command Classes MUST still be supported when a node is running in restricted mode:

- Anti-Theft Command Class
- Anti-Theft Unlock Command Class
- Manufacturer Specific Command Class
- Version Command Class
- Wake-Up Command Class
- All Command Classes that MUST always be supported at the non-secure level, refer to SDS13548 (see “This Command Class MUST always be in the NIF if supported”)

All other supported command classes SHOULD be removed in the restricted mode.

The locked state may be changed back to unlocked with this Command Class or the Anti-Theft Unlock Command Class.

When in restricted mode, a node MUST still be able to enter learn mode and join or leave a Z-Wave Network and support Z-Wave protocol operations.

4.8.1.3 Multi Channel Considerations

Multi Channel End Points SHOULD NOT support Anti-theft Command Class.

4.8.2 Anti-Theft Set Command

This command is used to lock or unlock a node.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ANTITHEFT (0x5D)							
Command = ANTITHEFT_SET (0x01)							
State	Number of Magic Code bytes						
Magic Code 1							
…							
Magic Code N							
Manufacturer ID (MSB)							
Manufacturer ID (LSB)							
Anti-theft Hint Length							
Anti-theft Hint							
…							
Anti-theft Hint M							
Z-Wave Alliance locking entity ID (MSB)							
Z-Wave Alliance locking entity ID (LSB)							

State (1 bit)

This field MUST indicate the desired locked/unlocked state for the receiving node.

The value 0 MUST indicate that the supporting node MUST change its state to unlocked.
The value 1 MUST indicate that the supporting node MUST change its state to locked.

If the receiving node is currently locked and this field is set to 0 (unlock), the receiving node MUST change its state to unlocked if the *Magic Code* field value matches the Magic Code value that was used to lock the node.

If the receiving node is currently locked and this field is set to 1 (lock), the receiving node MUST ignore this command.
If the receiving node is currently unlocked and this field is set to 0 (unlock), the receiving node MUST ignore this command.

If the receiving node is currently unlocked and this field is set to 1 (lock), the receiving node MUST change its state to lock and save the associated Magic code, Manufacturer ID, Anti-Theft Hint and Z-Wave Alliance locking entity ID.

Magic Code Length (7 bit)

This field MUST indicate the length (in byte) of the *Magic Code* field.
This field MUST be in the range 1..10.

Magic Code (N bytes)

This field contains the Magic Code used to lock or unlock the node.

The length of this field in byte MUST be according to the *Magic Code Length* field.

The receiving node MUST change its state from locked to unlocked if and only if this field's value matches the Magic Code that was used to lock the node.

Manufacturer ID (2 bytes)

This field describes the Z-Wave Manufacturer ID of the company's product that has locked the node.

This field MUST be set to 0x00 by a sending node and ignored by a receiving node if the S*tate* field is set to 0.

If a controlling node has a Z-Wave Manufacturer ID assigned, this field MUST be set to its assigned Manufacturer ID in SDS13425.

If a controlling node has no Z-Wave Manufacturer ID assigned, this field MUST be set to 0x00.

Anti-theft Hint Length (8 bits)

This field MUST indicate the length (in byte) of the *Anti-Theft Hint* field.
This field MUST be in the range 0..10.

Anti-theft Hint (N bytes)

This field is used as an identifier or key value to help retriving the Magic Code.

The length of this field in byte MUST be according to the *Anti-Theft Hint Length* field. This field MUST be omitted if the *Anti-Theft Hint Length* field is set to 0.

The format of this field is manufacturer/controlling node specific.

Z-Wave Alliance locking entity ID (2 bytes)

This field MUST specify a unique identifier for the entity that has locked the node.

A supporting node MUST NOT change its state to locked if this field is omitted or set to 0x00.
A controlling node MUST NOT lock a device without having a valid Z-Wave Alliance locking entity ID value.

Contact the Z-Wave Alliance to get an ID assigned. With this ID, the Z-Wave alliance will be able to provide the following information:

- Name of the organization
- Contact Phone for unlock information
- Optional website/url
- Optional unlock support email address

This field MUST be set to 0x00 if the *state* field is set to 0.
This field MUST NOT be set to 0x00 if the *state* field is set to 1.

4.8.3 Anti-theft Get Command

This command is used to request the locked/unlocked state of a supporting node.

The Anti-Theft Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ANTITHEFT (0x5D)							
Command = ANTITHEFT_GET (0x02)							

4.8.4 Anti-Theft Report Command

This command is used to advertise the lock/unlock state of a supporting node.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ANTITHEFT (0x5D)							
Command = ANTITHEFT_REPORT (0x03)							
Anti-theft Protection Status							
Manufacturer ID MSB							
Manufacturer ID LSB							
Anti-theft Hint Length							
Anti-theft Hint 1							
…							
Anti-theft Hint N							
Z-Wave Alliance locking entity ID (MSB)							
Z-Wave Alliance locking entity ID (MSB)							

Anti-theft Protection Status (8 bits)

Anti-theft Protection Status specifies the actual status of Z-Wave device in question. Refer to the table below with respect to defined status values.

Value	Description
0x00	Reserved.
0x01	Anti-Theft Protection is currently disabled, the node is unlocked
0x02	Anti-Theft Protection is currently enabled, the node is locked. However, it did not change network so it is fully functional.
0x03	Anti-Theft Protection is currently enabled, the node is locked. The node was reset and/or changed network and runs in restricted mode.
0x04..0xFF	Reserved

Manufacturer ID (2 bytes)

This field describes the Z-Wave Manufacturer ID of the company's product that has locked the node.

This field MUST be set to 0x00 if the *Anti-theft Protection Status* field is set to 0x01.

If the *Anti-theft Protection Status* field is set to 0x02 or 0x03 and a Z-Wave Manufacturer listed in [12] has locked the node, this field will advertise the Manufacturer ID that was provided in the Anti-theft Set Command.

If the *Anti-theft Protection Status* field is set to 0x02 or 0x03 and an entity who's not a Z-Wave Manfacturer listed in [12] has locked the node, this field will be set to 0x00.

Anti-theft Hint Length (8 bits)

This field MUST indicate the length (in byte) of the *Anti-Theft Hint* field.
This field MUST be in the range 0..10.

If the *Anti-theft Protection Status* field is set to 0x01, a sending node MUST set this field to 0.
If the *Anti-theft Protection Status* field is set to 0x02 or 0x03, a sending node MUST advertise the length of the Anti-Theft Hint field that was used to lock the node.

Anti-theft Hint (N bytes)

This field is used as an identifier or key value to help retrieving the Magic Code.

The length of this field in byte MUST be according to the *Anti-Theft Hint Length* field. This field MUST be omitted if the *Anti-Theft Hint Length* field is set to 0.

A supporting node MUST advertise the value that was specified in the Anti-Theft Set Command when it got locked.

Z-Wave Alliance locking entity ID (2 bytes)

This field advertises a unique identifier for the entity that has locked the node. SDS14622 contains the valid assigned IDs with additional information about the entity.

Contact the Z-Wave Alliance to get an ID assigned. With this ID, the Z-Wave alliance will be able to provide the following information:

- Name of the organization
- Contact phone for unlock information
- Optional website/url
- Optional unlock support email address

This field MUST be set to 0x00 if the *Anti-theft Protection Status* field is set to 0x01.

4.9 Anti-theft Unlock Command Class, version 1

This Command Class is used to unlock a device that has been locked by the Anti-theft Command Class. Controllers that do not lock devices still can control this command class in order to unlock devices that have been locked by previous owners.

4.9.1 Compatibility Considerations

A node supporting this Command Class MUST also support the Anti-Theft Command Class, version 3.

4.9.1.1 Multi Channel Considerations

Multi Channel End Points SHOULD NOT support the Anti-Theft Unlock Command Class.

4.9.2 Anti-Theft Unlock State Get Command

This command is used to request the locked/unlocked state of the node.

The Anti-Theft Unlock State Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ANTITHEFT_UNLOCK (0x7E)							
Command = COMMAND_ANTITHEFT_UNLOCK_STATE_GET (0x01)							

4.9.3 Anti-Theft Unlock State Report Command

This command is used to advertise the current locked/unlocked state of the node with some additional information.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ANTITHEFT_UNLOCK (0x7E)							
Command = COMMAND_ANTITHEFT_UNLOCK_STATE_REPORT (0x02)							
Reserved		Anti-theft Hint Length				Restricted	State
Anti-theft Hint							
…							
Anti-theft Hint N							
Manufacturer ID (MSB)							
Manufacturer ID (LSB)							
Z-Wave Alliance locking entity ID (MSB)							
Z-Wave Alliance locking entity ID (MSB)							

State (1 bit)

This field MUST indicate the current locked/unlocked state of the device.

The value 0 MUST indicate that the supporting node is unlocked.
The value 1 MUST indicate that the supporting node is locked.

The value 0x01 from the *Anti-theft Protection Status* field from the Anti-theft Report Command MUST be mapped to the value 0 for this field

The values 0x02 and 0x03 from the *Anti-theft Protection Status* field from the Anti-theft Report Command MUST be mapped to the value 1 for this field

Restricted (1 bit)

This field MUST indicate if the node currently runs in restricted mode.

The value 0 MUST indicate that the supporting node is not restricted.
The value 1 MUST indicate that the supporting node is restricted.

This field MUST be set to 0 if the *State* field is set to 0.

The values 0x01 and 0x02 from the *Anti-theft Protection Status* field from the Anti-theft Report Command MUST be mapped to the value 0 for this field

The value 0x03 from the *Anti-theft Protection Status* field from the Anti-theft Report Command MUST be mapped to the value 1 for this field

Anti-theft Hint Length (4 bits)

This field MUST indicate the length (in byte) of the *Anti-Theft Hint* field.
This field MUST be in the range 0..10.

If the state field is set to 0, a sending node MUST set this field to 0.
If the state field is set to 1, a sending node MUST advertise the length of the Anti-Theft Hint field that was used to lock the node.

Anti-theft Hint (N bytes)

This field is used as an identifier or key value to help retrieving the Magic Code.

The length of this field in byte MUST be according to the *Anti-Theft Hint Length* field. This field MUST be omitted if the *Anti-Theft Hint Length* field is set to 0.

A supporting node MUST advertise the value that was specified in the Anti-Theft Set Command when it got locked.

Manufacturer ID (2 bytes)

This field describes the Z-Wave Manufacturer ID of the company's product that has locked the node.

This field MUST be set to 0x00 if the state field is set to 0.

If the state field is set to 1 and a Z-Wave Manufacturer listed in [12] has locked the node, this field will advertise the Manufacturer ID that was provided in the Anti-theft Set Command.

If the state field is set to 1 and an entity who's not a Z-Wave Manfacturer listed in [12] has locked the node, this field will be set to 0x00.

Z-Wave Alliance locking entity ID (2 bytes)

This field advertises a unique identifier for the entity that has locked the node. SDS14622 contains the valid assigned IDs with additional information about the entity.

Contact the Z-Wave Alliance to get an ID assigned. With this ID, the Z-Wave alliance will be able to provide the following information:

- Name of the organization
- Contact phone for unlock information
- Optional website/url
- Optional unlock support email address

This field MUST be set to 0x00 if the state field is set to 0.

If the state field is set to 1, this field MUST advertise the ID that was provided in the Anti-Theft Set Command.

4.9.4 Anti-Theft Unlock Set Command

This command is used to unlock a node that is currently locked.

A receiving node MUST ignore this command if its current state is unlocked.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ANTITHEFT_UNLOCK (0x7E)							
Command = COMMAND_ANTITHEFT_UNLOCK_SET (0x03)							
Reserved				Magic code length			
Magic Code 1							
…							
Magic Code N							

Magic Code Length (4 bits)

This field MUST indicate the length in byte of the *Magic Code* field.
This field MUST be in the range 1..10

Magic Code (N bytes)

This field contains the Magic Code used to unlock the node.

The length of this field in byte MUST be according to the *Magic Code Length* field.

The receiving node MUST change its state to unlocked (and exit restricted mode) if this field's value matches the Magic Code that was used to lock the node.

4.10 Authentication Command Class, version 1

Authentication Command Class is a generic command class which can be used for transporting different technology authentication data contents (e.g., RFID, Magnetic Cards, etc.) in a Z-Wave network for access control systems.

4.10.1 Terminology

Authentication Data contents are initialized on a supporting node using a unique Authentication Data ID. Each Authentication Data ID also has an Authentication Technology Type, representing the input or technology with which the Authentication Data can be used. A user having access with several types of input technologies will have a Data Identifier for each of the technologies.

A node may support one or more authentication technologies including User Code. Combination of valid sets of Authentication Data can be configured on a supporting node using a unique Authentication ID. The Authentication ID will for example allow a user only if he/she inputs both an RFID tag and the corresponding User Code. An actual user may be associated with several Authentication IDs.

An example of Authentication Data ID and Authentication ID configurations are described in Figure 6

A node may support a Checksum functionality. A controlling node can request a checksum representing all authentication data content (either Authentication IDs or Authentication Data IDs) defined at the supporting node to ensure that the authentication data content databases are synchronized.

4.10.2 Interoperability Considerations

This Command Class can be used in conjunction with the Entry Control Command Class, to report the Authentication Data input to a controlling application.

This Command Class can be used in conjunction with the Door Lock Command Class or Barrier Operator Command Class. In this case, a supporting node MUST reflect Authentication Data inputs in the door lock status when relevant. (e.g. when the door becomes unsecured by Authentication Data input, the Door Lock Operation mode is updated to unsecure with an optional timeout).

A node supporting this Command Class MAY have an Association group issuing the corresponding Door Lock Operation Set Commands or Basic Set Commands when there are a valid Authentication Data input in order to control other door locks.

A supporting node may support several authentication technologies such as User Code, RFID, Cards and etc. A combination these technologies may also be needed to get access. In this case, a supporting node can be configured to accept certain combinations of authentication technologies.

An assigned User Code in the User Code Command Class may be part of an Authentication ID. In this case, all the Authentication credentials (including the User Code) defined in the Authentication ID MUST be input at the node in order to trigger the outcome defined by the User Code Status.
In other words, the User Code input alone is no longer enough, it will require all the credentials defined in the Authentication ID to get access. An example is given in Figure 5

4.10.3 Authentication Capability Get Command

This command is used to request the Authentication technology capabilities, number of Authentication ID and Data ID supported by the receiving node.

The Authentication Capability Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_AUTHENTICATION (0xA1)							
Command = AUTHENTICATION_CAPABILITIES_GET (0x01)							

4.10.4 Authentication Capability Report Command

This command is used to advertise the Authentication Command Class capabilities of sending node.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_AUTHENTICATION (0xA1)							
Command = AUTHENTICATION_CAPABILITIES_REPORT (0x02)							
Supported Data ID Entries (MSB)							
Supported Data ID Entries (LSB)							
Supported Authentication ID Entries (MSB)							
Supported Authentication ID Entries (LSB)							
MAR	MADR	OR	Reserved	Supported Authentication Technology Type Bit Mask length			
Supported Authentication Technology Type Bit Mask 1							
…							
Supported Authentication Technology Type Bit Mask N							
Supported Checksum Type Bit Mask							
Reserved				Supported Fallback Status Bit Mask length			
Supported Fallback Status Bit Mask 1							
…							
Supported Fallback Status Bit Mask N							

Reserved

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Supported Data ID Entries (16 bits)

This field is used to advertise the number of supported Authentication Data ID Entries.

The first byte MUST carry the most significant byte of the 16 bits value.

This field MUST be set to the total amount of supported Authentication Data IDs. This field MUST be in the range 1…65535.

Supported Authentication ID Entries (16 bits)

This field is used to advertise the number of supported Authentication ID Entries.

The first byte MUST carry the most significant byte of the 16 bits value.

This field MUST be set to the total amount of supported combinations entries. This field MUST be in the range 1…65535.

MAR (Multiple Authentication ID Report) Support (1 bit)

This field indicates if the sending node supports reporting the Multiple Authentication ID Blocks at once in a single Authentication Technologies Combination Report Command. This functionality should be supported by node supporting large amount of Authentication ID Entries.

The value 1 MUST indicate that the Multiple Authentication ID Report functionality is supported.
The value 0 MUST indicate that the Multiple Authentication ID Report functionality is not supported.

MADR (Multiple Authentication Data ID Report) Support (1 bit)

This field indicates if the sending node supports reporting the Multiple Authentication Data ID Blocks at once in a single Authentication Data Report Command. This functionality should be supported by node supporting large amount of Authentication Data ID Entries.

The value 1 MUST indicate that the Multiple Authentication Data ID Report functionality is supported.
The value 0 MUST indicate that the Multiple Authentication Data ID Report functionality is not supported.

OR Support (1 bit)

This field is used to advertise if the node supports being set Authentication ID combinations using a OR logic (User Code OR Authentication data ID 1 OR … OR Authentication data ID N)

The value 1 MUST indicate that the OR flag in the Authentication Technologies combination is supported.
The value 1 MUST indicate that the OR flag in the Authentication Technologies combination is not supported.

Supported Authentication Technology Type Bit Mask length (4 bits)

This field MUST advertise the length in bytes of the *Supported Authentication Technology Type Bit Mask* field carried in the command.

This field MUST be set to the minimum value which allows advertising all supported Authentication Technology types.

Supported Authentication Technology Type Bit Mask (N bytes)

This field advertises the supported Authentication Technology Type values. The length of this field in bytes MUST match the value advertised in the *Supported Authentication Technology Bit Mask Length* field.

Authentication Technology Type values are described in Table 13. In this field the supported Authentication Technology Type values MUST be encoded as follow:

- Bit 0 in Bit Mask 1 represents Authentication Technology Type 0x00 (reserved) and MUST be set to 0.
- Bit 1 in Bit Mask 1 represents Authentication Technology Type 0x01.
- Bit 2 in Bit Mask 1 represents Authentication Technology Type 0x02.
- …

If an Authentication Technology Type is supported, the corresponding bit MUST be set to ‘1’.
If an Authentication Technology Type is not supported, the corresponding bit MUST be set to ‘0’.

All other bits are reserved and MUST be set to zero by a sending node. Reserved bits MUST be ignored by a receiving node.

Supported Checksum Type Bit Mask (8 bits)

This field advertises the supported Checksum Type values.

Checksum Type values are described in Table 15. The supported Checksum Type values MUST be encoded as follow:

- Bit 0 in Bit Mask 1 represents Checksum Type 0x00.
- Bit 1 in Bit Mask 1 represents Checksum Type 0x01.
- …

If a Checksum Type is supported, the corresponding bit MUST be set to '1'.
If a Checksum Type is not supported, the corresponding bit MUST be set to '0'.

All other bits are reserved and MUST be set to zero by a sending node. Reserved bits MUST be ignored by a receiving node.

Supported Fallback Status Bit Mask length (4 bits)

This field MUST advertise the length in bytes of the *Supported Fallback Status Bit Mask* field carried in the command.

This field MUST be set to the minimum value which allows advertising all supported Fallback Statuses.

Supported Fallback Status Bit Mask (N bytes)

This field advertises the supported Fallback Status values. The length of this field in bytes MUST match the value advertised in the *Supported Fallback Status Bit Mask Length* field.

Fallback Status values are described in Table 14. This field MUST be encoded as follow:

- Bit 0 in Bit Mask 1 represents Fallback Status 0x00.
- Bit 1 in Bit Mask 1 represents Fallback Status 0x01.
- …

Fallback Statuses 0x00 and 0x01 MUST be supported by all supporting nodes.

If a Fallback Status is supported, the corresponding bit MUST be set to '1'.
If a Fallback Status is not supported, the corresponding bit MUST be set to '0'.

All other bits are reserved and MUST be set to zero by a sending node. Reserved bits MUST be ignored by a receiving node.

4.10.5 Authentication Data Set Command

This command is used to define Authentication Data for a given Data Identifier at the receiving node.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_AUTHENTICATION (0xA1)							
Command = AUTHENTICATION_DATA_SET (0x03)							
Authentication Data ID 1 (MSB)							
Authentication Data ID 2 (LSB)							
Reserved				Authentication Technology Type			
Authentication Data length							
Authentication Data 1							
…							
Authentication Data N							

Reserved

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Authentication Data ID (16 bits)

This field is used to identify an Authentication Data entry.

This field MUST be in the range 0..Max Number of Supported Data ID Entries advertised by the node in the Authentication Capability Report Command.

The first byte MUST carry the most significant byte of the 16 bits.

Values in the range 1..Max Number of Supported Data ID Entries MUST indicate the actual Authentication Data ID record to update.

The value 0 MUST indicate that the receiving node MUST erase the Authentication Data for all supported Authentication Data ID that are associated to specified *Authentication Technology Type*. If this field is set to 0x00, the Authentication Data length MUST be set to 0x00.

Authentication Technology Type (4 bits)

This field is used to specify the authentication medium associated to the Authentication Data ID. The field MUST be encoded according to Table 13.

Table 13, Authentication Data Set:: Authentication Technology Type encoding

Value	Authentication Technology	Version
0x00	Reserved	1
0x01	RFID tag	1
0x02	Magnetic card	1
0x03..0x0F	Reserved	1

Authentication Data length (8 bits)

This field is used to describe the length in bytes of the *Authentication Data* field.

The value 0 MUST indicate to erase the data entry represented by the Data Identifier.

Authentication Data (N bytes)

This field is used to advertise the Authentication Data contents to be set the specified Authentication Data ID.

The length of this field in bytes MUST be according to the corresponding *Authentication Data Length* field value. If the *Authentication Data Length* field is set to 0x00, this field MUST be omitted.

4.10.6 Authentication Data Get Command

This command is used to request the Authentication Data defined for a given Authentication Data ID.

The Authentication Data Report MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_AUTHENTICATION (0xA1)							
Command = AUTHENTICATION_DATA_GET (0x04)							
Authentication Data ID 1 (MSB)							
Authentication Data ID 2 (LSB)							
Reserved							Report More

Authentication Data ID (16 bits)

This field is used to describe the requested authentication Data Identifier.

The first byte MUST carry the most significant byte of the 16 bit value.

A supporting node MUST return this value in the *Authentication Data ID* field of the first Authentication Data ID Block in the returned Authentication Data Report Command.

Report More (1 bit)

This field is used to instruct the receiving node to report as many Authentication Data ID as possible within a single Z-Wave command (that fits in a single Z-Wave frame) because the sending node intends to read the whole (or a large part of the) Authentication Data database.

This field MUST be ignored by a receiving node if it advertises no support for Multiple Authentication Data Report (MADR) in the Authentication Capability Report Command.

The value 0 MUST indicate to return a report for the requested Authentication Data ID only.
The value 1 MUST indicate to return a report for the requested Authentication Data ID and additionally report as many Authentication Data ID blocks as possible in the response.

When this field is set to 1, a node advertising support for Multiple Authentication Data Report (MADR) in the Authentication Capability Report Command:

- SHOULD return at least 2 Authentication Data ID blocks in the Authentication Data Report Command unless the last used Authentication Data ID or a non-supported Authentication Data ID is requested.
- SHOULD return as many consecutive non-empty Authentication Data ID record as possible.
- The subsequent Authentication Data ID blocks MUST contain consecutive Authentication Data ID having Authentication Data length different than 0. For example, a node supports 5 authentication Data IDs and both RFID (0x01) and Magnetic card (0x02) authentication technologies, and the setting is defined as follow. In this case, the supporting node MUST report Authentication Data ID 1 and 4 when requested about Authentication Data ID 1.
    - Authentication Data ID 1, Technology Type = 0x01, Data length = 3, Data = C10001
    - Authentication Data ID 2, Technology Type = 0x01, Data length = 0
    - Authentication Data ID 3, Technology Type = 0x01, Data length = 0
    - Authentication Data ID 4, Technology Type = 0x02, Data length = 2, Data = CDF1
    - Authentication Data ID 5, Technology Type = 0x02, Data length = 0

4.10.7 Authentication Data Report Command

This command is used to advertise the Authentication Data assigned at the supporting node for a given Authentication Data ID entry.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_AUTHENTICATION (0xA1)							
Command = AUTHENTICATION_DATA_REPORT (0x05)							
Number of Authentication Data ID Blocks							
Authentication Data ID 1 (MSB) 1							
Authentication Data ID 2 (LSB) 1							
Reserved				Authentication Technology Type 1			
Authentication Data Length 1							
Authentication Data 1 - 1							
…							
Authentication Data N - 1							
…							
Authentication Data ID 1 (MSB) M							
Authentication Data ID 2 (LSB) M							
Reserved				Authentication Technology Type M			
Authentication Data Length M							
Authentication Data 1 - M							
…							
Authentication Data N - M							
Next Authentication Data ID 1 (MSB)							
Next Authentication Data ID 2 (LSB)							

Number of Authentication Data ID Blocks (8 bits)

This field is used to specify how many Authentication Data ID blocks are contained in the actual command.

This field MUST be in the range 1..255. A node MUST respect the Z-Wave MAC frame size or Transport service limits when sending this command.

The number of Authentication Data ID contained in the command MUST be according to this field. An Authentication Data ID block MUST comprise the following fields:

- Authentication Data (16 bits)
- Authentication Technology Type (4 bits)
- Authentication Data Length (8 bits)
- Authentication Data (N bytes)

This field MUST be set to 1 by a node advertising no support for Multiple Authentication Data ID Block Report (MADR) in the User Code Capabilities Report Command.

Authentication Data ID (16 bits)

This field is used to describe the advertised authentication Data Identifier.

If this field is set to 0x00 or a non-supported Authentication Data ID, the *Authentication Technology Type* and the *Authentication Data Length* fields MUST be set to 0x00.

Authentication Technology Type (4 bits)

This field is used to advertise the authentication medium associated to the Authentication Data ID. The field MUST be encoded according to Table 13.

Authentication Data length (8 bits)

This field is used to describe the length in bytes of the *Authentication Data* field.

The value 0 MUST indicate that the specified Authentication Data ID is not supported.

Authentication Data (N bytes)

This field is used to advertise the Authentication Data contents set for the specified Authentication Data ID.

The length of this field in bytes MUST be according to the corresponding *Authentication Data Length* field value. If the *Authentication Data Length* field is set to 0x00, this field MUST be omitted.

Next Authentication Data ID (16 bits)

This field is used to advertise the next Authentication Data ID set at the sending node with non-empty Authentication Data.

This field MUST be set to 0x00 if there are no more Data Identifiers are defined after the Authentication Data ID value of the *Authentication Data ID* field of the last Authentication Data ID block.

4.10.8 Authentication Technologies Combination Set Command

This command is used to define the Combination Entries at the receiving node.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_AUTHENTICATION (0xA1)							
Command = AUTHENTICATION_TECHNOLOGIES_COMBINATION_SET (0x06)							
Authentication ID 1 (MSB)							
Authentication ID 2 (LSB)							
Fallback Status							
User Identifier 1 (MSB)							
User Identifier 2 (LSB)							
Schedule ID 1 (MSB)							
Schedule ID 2 (LSB)							
OR logic	Number of Authentication Data ID						
Authentication Data ID 1 (MSB) 1							
Authentication Data ID 2 (LSB) 1							
…							
Authentication Data ID 1 (MSB) M							
Authentication Data ID 2 (LSB) M							

Reserved

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Authentication ID (16 bits)

This field is used to specify the Authencation ID that is to be updated.

This field MUST be in the range 0..max number of Supported Authentication ID Entries

The first byte MUST carry the most significant byte of the 16 bits.

The value 0 is used to erase all the Authentication Data for all supported Authentication ID Entries. If this field is set to 0x00 and all the other fields MUST set to 0x00, and a receiving node MUST erase all configured Authentication ID Entries.

Values in the range 1..Max Number of Supported Authentication ID Entries MUST indicate the actual Authentication ID record to update.

Fallback Status (8 bits)

This field is used to configure the status of Authentication ID at the receiving node if a User Code entry is not defined (i.e. *User Identifier* field set to 0x00).

The status represents the outcome to perform at the node when all the Authentication Data defined in the Authentication ID has been input at the supporting node, during the defined Schedule ID.

This field MUST set but ignored if the User Identifier is set to a value greater than 0x00. Instead, the User ID Status defined for the corresponding User Identifier in the User Code Command Class MUST take precedence over this field.

If the *User Identifier* field is set to 0x00, this field MUST be encoded and interpreted according to Table 14.

Table 14, Authentication Technologies Combination Set:: Fallback Status encoding

Value	Description	Version
0x00	Available: The database entry represented by this Authentication ID is not in use. This status MUST be used to delete Authentication ID entries.	1
0x01	Enabled / Grant Access The database entry represented by this Authentication ID is in use and enabled. When a user has input all required Authentication credentials associated to this Authentication ID, the node: • MUST indicate that the input was accepted • MUST grant access to the user. • MUST secure the Door Lock again after a timeout (either configured if set in Timed Operation or pre-defined if the Door Lock is set in Constant Operation)	1
0x02	Disabled: The database entry represented by this Authentication ID is in use but disabled. E.g. It allows a controller to push Authentication ID combinations to a supporting node and let these credentials be subsequently activated by a local interface. When a user has input all required Authentication credentials associated to this Authentication ID, the node: • MUST indicate that the input was not accepted • MUST NOT grant access to the user.	1

Value	Description	Version
0x03	Messaging: The database entry represented by this Authentication ID is in use. The value is used to relay notifications to a controlling application. E.g. A user may enter a messaging code for notifying that maintenance started or an application should activate the alarm system. When a user has input all required Authentication credentials associated to this Authentication ID, the node: • SHOULD indicate that the input was accepted • MUST NOT grant access to this user • MUST NOT take preventive actions for further attempts to enter input. • MUST send an Entry Control Notification or a Notification Report via the Lifeline Group to identify the messaging User Identifier or Authentication ID.	1
0x04	Passage Mode: The database entry represented by this Authentication ID is in use. The value is used to let the Door Lock permanently grant access. Passage Mode can be activated and deactivated using the Door Lock Operation Set (Unsecured without timeout / Secured) or via a Passage Mode credential input. E.g. A receptionist comes in the morning and enters a Passage Mode credentials to allow anybody to subsequently enter the premises without entering special access codes until Passage Mode is deactivated again. When a user has input all required Authentication credentials associated to this Authentication ID, the node: • MUST indicate that the code is accepted • MUST toggle the Door Lock operation state between secured and unsecured without timeout (and also disable any auto-relock functionality). • MUST send a Door Lock Operation Report and Door Lock Configuration Report via the Lifeline Association Group to advertise the new Door Lock state.	1
0x05..0xFF	Reserved	1

User Identifier (16 bits)

This field is used to specify the identity of the User Code (User Code Command Class) which MUST be used in combination with authentication data to get access.

If User Identifier has a User ID Status set to 0x00 in the User Code Command Class, the combination will not allow access as it cannot be input at the supporting node.

This field is set to a value larger than 0x00 and the supporting node does not support the User Code Command Class, the Authentication ID will be ignored as such a User Code cannot be input at the supporting node.

If a User Code is defined in an Authentication ID with an AND logic, the input of the User Code alone is no longer sufficient to trigger the outcome defined by the User ID Status. All Authentication Data indicated by the Authentication Data IDs in this command MUST be input to trigger the outcome specified by the User ID Status.

Schedule ID (16 bits)

This field is used specify the Schedule during which the Authentication ID is active.

The value 0x00 MUST indicate that the Authentication ID is active at all times.

Values greater than 0 MUST indicate that the Authentication ID is valid or active only during the corresponding Schedule ID, defined in the Generic Schedule Command Class.

OR logic (1 bit)

This field is used to specify if OR logic is to be applied for the actual Authentication ID.

Nodes advertising no support for the OR in the Authentication Capability Report Command MUST ignore this field and apply an AND logic.

If this field is set to 0, an AND logic MUST be applied for the actual Authentication ID.
In this case, the User Code (if defined) and all the defined Authentication Data must be input (during the Schedule ID) in order to get access or provide the outcome defined in the corresponding User Code Status / Fallback Status.

If this field is set to 1, an OR logic MUST be applied for the actual Authentication ID.
In this case, the User Code (if defined) or any of the defined Authentication Data must be input (during the Schedule ID) in order to get access or provide the outcome defined in the corresponding User Code Status / Fallback Status.

Number of Authentication Data ID (7 bits)

This field is used to specify how many Authentication Data ID are contained in the the actual Authentication ID. Each Authentication Data ID MUST be 16 bits long.

This field MUST be set to 0, if no Authentication Data ID MUST be used for the combination and the User Identifer (User Code) alone is enough to get access.

If this field is set to 0, the Authentication Data ID field MUST be omitted.

For example, if this field is set to 3, the corresponding Authentication Data ID field must be 6 bytes long.

Authentication Data ID (M x 16 bits)

This field is used to identify the list of Authentication Data ID that are required for the specified Authentication ID.

The first byte MUST carry the most significant byte of the 16 bits.

In an Authentication ID, if any of the Authentication Data ID does not have any Authentication Data defined (Authentication Data Length = 0), the Authentication ID MUST NOT permit any access.

4.10.9 Authentication Technologies Combination Get Command

This command is used for requesting defined combination entries of a specific Authentication ID.

The Authentication Technologies Combination Report MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_AUTHENTICATION (0xA1)							
Command = AUTHENTICATION_TECHNOLOGIES_COMBINATION_GET (0x07)							
Authentication ID (MSB)							
Authentication ID (LSB)							
Reserved							Report More

Authentication ID (16 bits)

This field is used to describe the requested Authentication ID.

The first byte MUST carry the most significant byte of the 16 bits.

A supporting node MUST return this value in the *Authentication Data ID* field of the first Authentication Data ID Block in the returned Authentication Technologies Combination Report Command.

Report More (1 bit)

This field is used to instruct the receiving node to report as many Authentication ID as possible within a single Z-Wave command (that fits in a single Z-Wave frame) because the sending node intends to read the whole (or a large part of the) Authentication ID database.

This field MUST be ignored by a receiving node if it advertises no support for Multiple Authentication Report (MAR) in the Authentication Capability Report Command.

The value 0 MUST indicate to return a report for the requested Authentication ID only.
The value 1 MUST indicate to return a report for the requested Authentication ID and additionally report as many Authentication ID blocks as possible in the response.

When this field is set to 1, a node advertising support for Multiple Authentication Report (MAR) in the Authentication Capability Report Command:

- SHOULD return at least 2 Authentication ID blocks in the Authentication Technolofies Combination Report Command unless the last used Authentication ID or a non-supported Authentication ID is requested.
- SHOULD return as many consecutive non-emtpy Authentication ID record as possible.
- The subsequent Authentication ID blocks MUST contain consecutive Authentication ID having User Identifier or Fallback Status different than 0.

4.10.10 Authentication Technologies Combination Report Command

This command is used to advertise defined Combination Entries of a specific Authentication ID.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_AUTHENTICATION (0xA1)							
Command = AUTHENTICATION_TECHNOLOGIES_COMBINATION_REPORT (0x08)							
Number of Authentication ID blocks							
Authentication ID block length 1							
Authentication ID 1 (MSB) 1							
Authentication ID 2 (LSB) 1							
Fallback Status 1							
User Identifier 1 (MSB) 1							
User Identifier 2 (LSB) 1							
Schedule ID 1 (MSB) 1							
Schedule ID 2 (LSB) 1							
OR logic 1	Number of Authentication Data IDs 1						
Authentication Data ID 1 (MSB) 1 -1							
Authentication Data ID 2 (LSB) 1 - 1							
…							
Authentication Data ID 1 (MSB) M - 1							
Authentication Data ID 2 (LSB) M- 1							
…							
Authentication ID block length N							
Authentication ID 1 (MSB) N							
Authentication ID 2 (LSB) N							
Fallback Status N							
User Identifier 1 (MSB) N							
User Identifier 2 (LSB) N							
Schedule ID 1 (MSB) N							
Schedule ID 2 (LSB) N							
OR logic N	Number of Authentication Data IDs N						
Authentication Data ID 1 (MSB) 1 - N							
Authentication Data ID 2 (LSB) 1–– N							
…							
Authentication Data ID 1 (MSB) M - N							
Authentication Data ID 2 (LSB) M- N							
Next Authentication ID 1 (MSB)							
Next Authentication ID 2 (LSB)							

Number of Authentication ID Blocks (8 bits)

This field is used to specify how many Authentication ID blocks are contained in the actual command.

This field MUST be in the range 1..255. A node MUST respect the Z-Wave MAC frame size or Transport service limits when sending this command.

The number of Authentication ID contained in the command MUST be according to this field. An Authentication Data ID block MUST comprise the following fields:

- Authentication ID Block Length (8 bits)
- Authentication ID (16 bits)
- Fallback Status (8 bits)
- User Identifier (16 bits)
- Schedule ID (16 bits)
- OR logic / Number of Authentication Data IDs (8 bits)
- Authentication Data ID (M x 16 bits)

This field MUST be set to 1 by a node advertising no support for Multiple Authentication ID Block Report (MAR) in the Authentication Capability Report Command.

Authentication ID block length (8 bits)

This field is used to indicate the total length of the actual Authentication ID Block. This field MUST advertise the length in bytes of the Authentication ID Block, including itself.

Authentication ID (16 bits)

This field is used to specify the Authentication ID for the actual Authentication ID Block.
The first byte MUST carry the most significant byte of the 16-bits value.

If this field is set to 0x00 or a non-supported Authentication ID, the *Fallback Status, User Identifier, Schedule ID and Number of Authentication Data ID* fields MUST be set to 0x00.

Fallback Status (8 bits)

This field is used to advertise the status of Authentication ID defined at the receiving node.

The status represents the outcome to perform at the node when all the Authentication Data defined in the Authentication ID block has been input at the supporting node, during the defined Schedule ID.

This field MUST be ignored if the User Identifier field in the corresponding Authentication ID Block is set to a value greater than 0x00. Instead, the User ID Status defined for the corresponding User Identifier in the User Code Command Class MUST take precedence over this field.

If the *User Identifier* field is set to 0x00, this field MUST be encoded and interpreted according to.

User Identifier (16 bits)

This field is used to advertise the identity of the User Code (User Code Command Class) which is used in combination with authentication data to get access for this Authentication ID block.

If User Identifier has a User ID Status set to 0x00 in the User Code Command Class, the the combination will not allow access as it cannot be input at the supporting node.

This field is set to a value larger than 0x00 and the supporting node does not support the User Code Command Class, the Authentication ID will be ignored as such a User Code cannot be input at the supporting node.

If a User Code is defined in an Authentication ID, the input of the User Code alone is no longer sufficient to trigger the outcome defined by the User ID Status. All Authentication Data indicated by the Authentication Data IDs in this command MUST be input to trigger the outcome specified by the User ID Status.

Schedule ID (16 bits)

This field is used specify the Schedule during which the Authentication ID is active.

The value 0 MUST indicate that the Authentication ID is active at all times.

Values greater than 0 MUST indicate that the Authentication ID is valid or active only during the corresponding Schedule ID, defined in the Generic Schedule Command Class.

OR logic (1 bit)

This field is used to advertise if OR logic is to be applied for the actual Authentication ID.

Nodes advertising no support for the OR in the Authentication Capability Report Command MUST ignore this field and set this field to 0.

If this field is set to 0, an AND logic MUST be applied for the actual Authentication ID block.
If this field is set to 1, an OR logic MUST be applied for the actual Authentication ID block.

Number Of Authentication Data IDs (7 bits)

This field is used to specify how many Authentication Data ID are contained in the the corresponding Authentication ID Block. Each Authentication Data ID MUST be 16 bits long.

For example, if this field is set to 3, the corresponding Authentication Data ID field must be 6 bytes long.

Authentication Data ID (M x 16 bits)

This field is used to identify the list of Authentication Data ID entries that compose the specified Authentication ID.

The length of this field MUST be according to the *Number Of Authentication Data IDs* field in the corresponding Authentication ID Block.

The first byte MUST carry the most significant byte of the 16 bits.

If any of the Authentication Data ID does not have any Authentication Data defined, the Authentication ID MUST NOT permit any access.

Next Authentication ID (16 bits)

This field is used to advertise the next Authentication ID set at the sending node with non-empty Authentication ID data.

This field MUST be set to the next Authentication ID which has a Fallback or User Identifier status is different from 0.

This field MUST be set to 0x00 if there are no more Authentication IDs are defined after the value of the *Authentication ID* field of the last Authentication ID Block.

4.10.11 Authentication Checksum Get Command

This command is used to request checksum(s) representing the current authentication data or combinations set at the supporting node.

A supporting node MUST return a response with the *Checksum* field set to 0x00 if the checksum type or the authentication type is not supported.

The Authentication Checksum Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_AUTHENTICATION (0xA1)							
Command = AUTHENTICATION_CHECKSUM_GET (0x09)							
Res	Checksum type			Authentication Technology Type			

Reserved

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Checksum Type (3 bits)

This field is used to indicate which checksum to return. This field MUST be encoded according to Table 15.

Table 15, Authentication Checksum Get:: Checksum type encoding

Value	Checksum Type	Version
0x00	Authentication Data checksum	1
0x01	Combination Entries checksum	1
0x02..0x07	Reserved	1

Authentication Technology Type (4 bits)

This field is used to specify which type of the authentication technology to use for the checksum calculation. The field MUST be encoded according to Table 13.

This field MUST be set to 0 by a sending node and ignored by a supporting node if the *Checksum Type* field is set to 1.

4.10.12 Authentication Checksum Report Command

This command is used to advertise the checksum representing current authentication data that correspond to a specific technology type or combination entries defined at the supporting node

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_AUTHENTICATION (0xA1)							
Command = AUTHENTICATION_DATA_CHECKSUM_REPORT(0x0F)							
Res	Checksum type			Authentication Technology Type			
Checksum 1 (MSB)							
Checksum 2 (LSB)							

Authentication Technology Type (4 bits)

This field is used to advertise which type of the authentication technology has been used for the checksum calculation. The field MUST be encoded according to Table 13.

This field MUST be set to 0x00 by the sending node if the Checksum type is set to 0x01.

The other values are reserved and MUST NOT be used by sending node. Reserved values MUST be ignored by a receiving node.

Checksum Type (3 bits)

This field is used to indicate which checksum to return. This field MUST be encoded according to Table 15.

Authentication Data Checksum (16 bits)

This field is used to advertise the checksum value. The checksum MUST be calculated using the CRC-CCITT polynomial using initialization value equal to 0x1D0F and 0x1021 (normal representation).

This field MUST be set to 0x0000 if no authentication data or combination entries are defined at sending node.

The calculation of the checksum depends on the *Checksum Type* and *Authentication Technology Type* field values.

If the checksum type field is set to 0x00 Authentication Data checksum:

Similar to the User Code checksum data format (refer to User Code Command Class, version 2), the Authentication Data checksum data MUST be built by concatenating initialized Authentication Data content for a given technology type (with Authentication Data length different than 0x00) in the ascending Data ID order.

If the Authentication Technology Type is set to 0x00, the checksum field MUST be set to 0x00.

Else, for example for RFID based authentication technology, each initialized RFID data MUST be formatted as follows:

Data ID (16 bits) | RFID Data (1..32 bytes)

For example, a node supporting 4 RFID IDs that are set as follow:

- Data ID 1, entry Data Content length 4, RFID code ABC10001
- Data ID 2, entry Data Content length 0
- Data ID 3, entry Data Content length 5, RFID code AB58198301
- Data ID 4, entry Data Content length 0

In this case, the defined RFID data content for Data ID 1 and 3 MUST be concatenated to obtain the checksum data:

0x0001 | 0xAB C1 00 01| 0x0003 |0xAB 58 19 83 01

The checksum data MUST be: 0x0001ABC100010003AB58198301 and the checksum field MUST be set to 0x17FE.

If the checksum type field is set to 0x01 Combination Entries checksum:

The checksum data MUST be built by concatenating defined Combination Entries whose status or User Identifier is different than 0x00 in the ascending Authentication ID order. Each entry MUST be comprised of all the fields included in an Authentication ID Block for the version of the supporting node, including reserved fields.

For example, a node supports both User Code and RFID based authentication technology, the Combination Entries checksum calculation MUST be formatted as follows:

- Authentication ID Block Length (8 bits)
- Authentication ID (16 bits)
- Fallback Status (8 bits)
- User Identifier (16 bits)
- Schedule ID (16 bits)
- Number of Authentication Data IDs (8 bits)
- Authentication Data ID (M x 16 bits)

Authentication ID Block Length (8 bits) |Authentication ID (16 bits) | Fallback Status (8 bits) |User Identifier (16 bits) |Schedule ID (16 bits) | Number of Authentication Data IDs (8 bits) | Authentication Data ID (M x 16 bits)

For example, a node supporting 3 Authentication IDs that are set as follow:

- Length 11, Authentication ID 1, Fallback status 0, User Identifier 1, Schedule ID 0, Number of Authentication Data IDs 1, Data Identifier 1.
- Length 9, Authentication ID 2, Fallback status 0, User Identifier 0, Schedule ID 0, Number of Authentication Data IDs 0.
- Length 13, Authentication ID 3, Fallback status 0, User Identifier 200, Schedule ID 1, Number of Authentication Data IDs 2, Data Identifier 4 and 5

In this case, the defined Combination Entries checksum data for Authentication ID 1 and 3 MUST be concatenated to obtain the checkum data:

0x0B | 0x0001 | 0x00 | 0x0001 | 0x0000 | 0x01 | 0x0001 | 0x0D | 0x0003 | 0x00 | 0x00C8 | 0x0001 | 0x02 | 0x0004 | 0x0005

The checksum data MUST be: 0x0B000100000100000100010D00030000C800010200040005 and the checksum must be 0x0A53.

4.10.13 Examples and use-cases

An example of configurations with User Code CC and RFID tags is shown in Figure 5

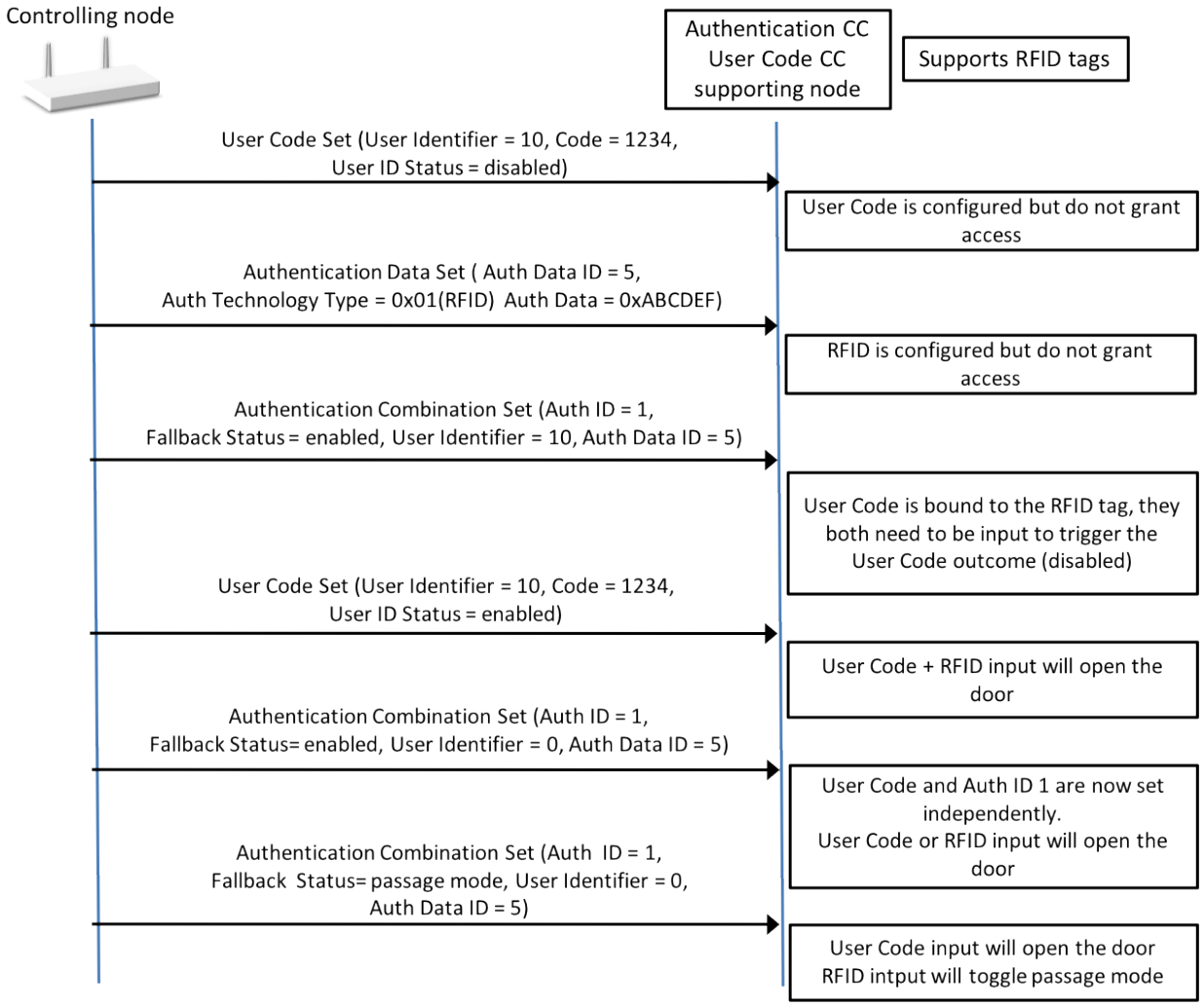


Figure 5, Authentication Command Class example: User Code and RFID configuration

An example of Authentication Data configurations with RFID and Magnetic card technology types is shown in Figure 6.

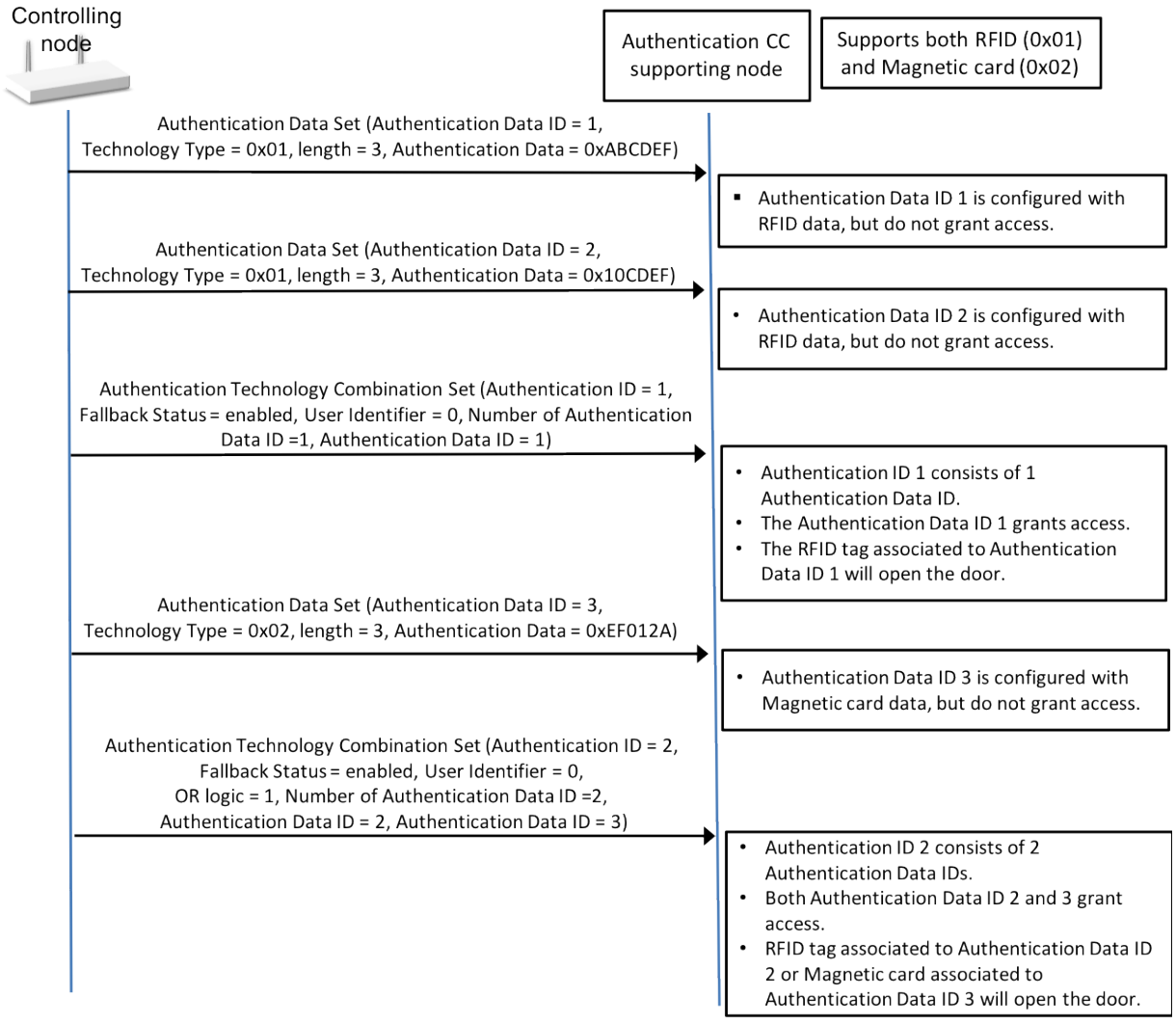


Figure 6: Authentication Command Class example: Authentication Data configuration

4.11 Authentication Media Write Command Class, version 1

This command class is used to write raw Authentication Media data, such as RFID data contents on tags.

The supported authentication medium itself is specific to individual nodes. Supporting nodes can use the Entry Control Command Class to report the corresponding readings of their supported Authentication Media.

4.11.1 Authentication Media Capability Get Command

This command is used to request the Authentication Media capabilities supported by the receiving node.

The Authentication Media Capability Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_AUTHENTICATION_MEDIA_WRITE (0xA2)							
Command = AUTHENTICATION_MEDIA_CAPABILITIES_GET(0x01)							

4.11.2 Authentication Media Capability Report Command

This command is used to advertise the Authentication Media Command Class capabilities supported by the sending node.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_AUTHENTICATION_MEDIA_WRITE (0xA2)							
Command = AUTHENTICATION_MEDIA_CAPABILITIES_REPORT (0x02)							
Reserved							DGS

DGS (Data Generation support) (1 bit)

This field is used to advertise if the supporting node has the capability to generate authentication data that can be written on its authentication media.

This feature is used if the authentication media has a particular format and would not accept any arbitral hexadecimal data.

If this field is set to 1, the supporting node MUST support creating authentication data for their write operations.
If this field is set to 0, the supporting node MUST NOT support creating authentication data for their write operations.

4.11.3 Authentication Media Write Start Command

This command is used to write Authentication Media contents (i.e., RFID data to a tag).

The Authentication Media Write Status Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_AUTHENTICATION_MEDIA_WRITE (0xA2)							
Command = AUTHENTICATION_MEDIA_WRITE_START (0x03)							
Sequence number							
Timeout							
Generate-data	Reserved						
Data Length							
Data 1							
…							
Data N							

Sequence number (8 bits)

This field is used to keep track of the Authentication Media Write Operation session.

A supporting node MUST return a Authentication Media Write Status Command with the status field set to 0x03 if receiving this command with a different sequence number than the ongoing write operation.

Timeout (8 bits)

This field is used to specify how many seconds the receiving node MUST wait between the reception of this command and aborting the write operation.

For nodes with a sensing/feeback capability for their authentication media write operation, this value MUST also be used to wait that a media is accessible to the receiving node before the write operation is aborted.

For Nodes without any feedback capability for their authentication media write operation MUST ignore this timeout value after and try to write the data on the authentication media target immediately.

Generate data (1 bit)

This field is used to indicate the supporting node to generate the Authentication data that will be written on the Authentication Media.

If this field is set to 1, the *Data Length* field MUST be set to 0.

If this field is set to 1, the supporting node MUST generate valid authentication data and return the written data in the *Data* field of the returned response.

Data Length (8 bits)

This field is used to describe the length in bytes of the corresponding *Data* field.

This field MUST be in the range 0..255.

This field MUST be set to a value higher than 0 if the *Generate Data* field is set to 0.

Data (N bytes)

This field is used to advertise the Data contents that MUST be written to the media.

The length of this field in bytes MUST be according to the corresponding *Data Length* field value.

4.11.4 Authentication Media Write Stop Command

This command is used to interrupt an ongoing Authentication Media Write operation.

The Authentication Media Write Status Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_AUTHENTICATION_MEDIA_WRITE (0xA2)							
Command = AUTHENTICATION_MEDIA_WRITE_STOP (0x04)							

4.11.5 Authentication Media Write Status Command

This command is used to advertise the status of an Authentication Media Write operation.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_AUTHENTICATION_MEDIA_WRITE (0xA2)							
Command = AUTHENTICATION_MEDIA_WRITE_STATUS (0x05)							
Sequence number							
Status							
Data Length							
Data 1							
…							
Data N							

Sequence number (8 bits)

This field is used to advertise the sequence number of the last Write operation.

If an Authentication Media Write Stop Command was received while no operation was ongoing, this field SHOULD be set to 0x00.

Data Length (8 bits)

This field is used to advertise the length in bytes of the *Data* field.

This field MUST be in the range 0..255.

This field MUST be set to a value highest than 0 if the *Status* field is set to 0xFE or 0xFF.

Data (N bytes)

This field is used to advertise the data that was written on the Authentication media as part of the write operation.

This field MUST contain the data that was written on the media if the *Status* field is set to 0xFE or 0xFF.

Status (8 bit)

This field is used to indicate the status of the Authentication Media Write operation. This field MUST encoded according to Table 63.

Table 16,Authentication Media Write Status::Status encoding

Value	Description
0x00	Failed: Write operation not supported The node does not support write operations.
0x01	Failed: Forced stop The node has received a Authentication Media Write Stop Command before the data contents could be written on the target.
0x02	Failed: timeout The node has timed out and could not write the data content on a target
0x03	Failed: Another operation is ongoing The node has did not initiate the write operation as another operation was already ongoing.
0x04	Failed: Aborted The node has aborted the operation due to local input or other higher priority task.
0x05	Failed: Buffer overflow The node could not start or aborted the write operation as it ran into a buffer overflow while caching the data to write.
0x06	Failed: Proprietary hex data restriction The node has received hexadecimal data that does not comply with the required format for its Authentication Media data. The controlling node should use the “Generate-data” flag instead. This status MUST NOT be returned by a node if it does not advertise support for Data Generation in the Authentication Media Capability Report Command
0x07	Failed: Non re-writable target The node could not write the data on the target as it already contains some data that could not be overwritten.
0xFE	Completed with no status: The Write Operation was carried out, but the node has no feedback from the media writing operation and cannot confirm that the data was properly written on the target.
0xFF	Success The specified data was properly written on the target.

4.12 Barrier Operator Command Class, version 1

The Barrier Operator Command Class is used to control and query the status of motorized barriers.

4.12.1 Compatibility considerations

CC:0066.01.00.23.001 A supporting node MAY ignore a Barrier Operator Command if the requested operation violates operational limitations or safety regulations.

4.12.1.1 Node Information Frame (NIF)

CC:0066.01.00.21.001 The Barrier Operator Command Class MUST be supported only using secure communication (S0 and/or S2 Command Class).

CC:0066.01.00.21.003 A supporting node MUST NOT advertise the Barrier Operator Command Class in its NIF after network inclusion.

4.12.1.2 Command Class dependencies

CC:0066.01.00.21.002 A node supporting the Barrier Operator Command Class MUST support the Notification Command Class, version 4 or newer.

4.12.2 Barrier Operator Set Command

This command is used to initiate an unattended change in state of the barrier.

CC:0066.01.01.13.001 A supporting node MAY ignore this command if the requested operation violates operational limitations or safety regulations.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_BARRIER_OPERATOR							
Command = BARRIER_OPERATOR_SET							
Target Value							

Target Value (8 bits)

CC:0066.01.01.11.001 This field MUST specify the intended state of the device.

CC:0066.01.01.11.002 The encoding of this field MUST be according to Table 17.

Table 17, Barrier Set::Target Value

Target Value		Description	Version
0x00	CLOSE	Initiate unattended close	1
0xFF	OPEN	Initiate unattended open	1

CC:0066.01.01.11.003 All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

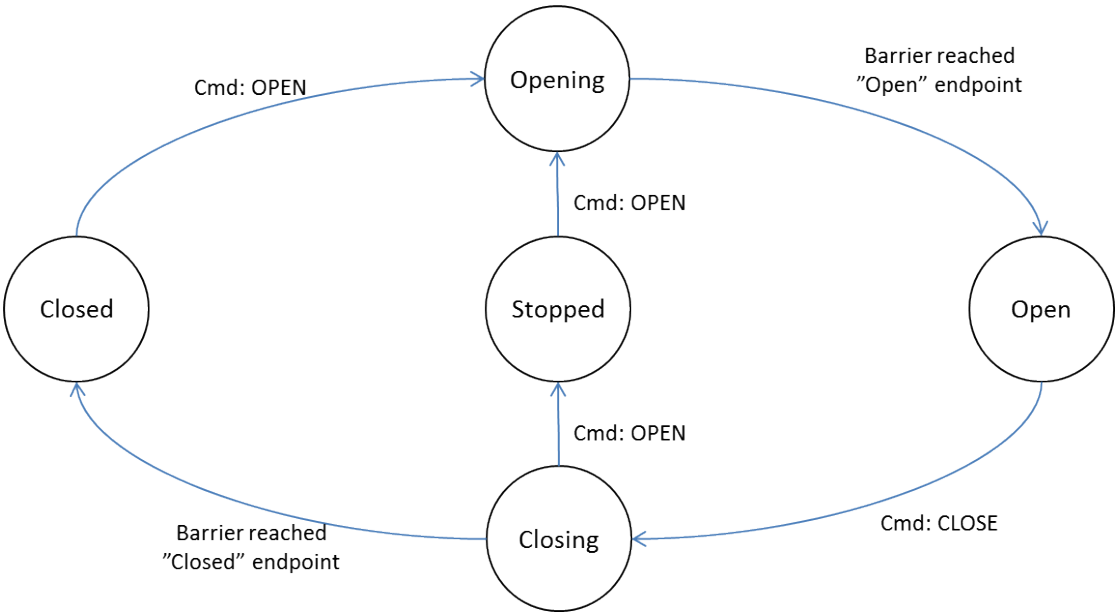


Figure 7, Barrier Set, state transitions

CC:0066.01.01.11.004 A receiving node MUST comply with the allowed state changes and triggering commands specified in Figure 7. Any state/command pair not specified in Figure 7 MUST NOT cause any action.

CC:0066.01.01.12.001
CC:0066.01.01.11.005 The receiving node SHOULD respond to OPEN and CLOSE commands. However, the receiving node MUST NOT respond to OPEN or CLOSE commands if any safety issue is detected.

CC:0066.01.01.11.006 The receiving node MUST NOT respond to the CLOSE command unless it is in the “Open” state.

4.12.2.1 Error Handling

CC:0066.01.01.11.007 The device MUST stop if any safety issue is detected.

CC:0066.01.01.11.008 If the requested operation is overruled by a safety mechanism, the device MUST notify the requester via a Notification Report with the Notification {Access Control::Barrier unable to perform requested operation due to UL requirements}.

CC:0066.01.01.11.009 If the requested operation is prohibited by the specific device class, e.g. SPECIFIC_TYPE_SECURE_BARRIER_OPEN_ONLY or SPECIFIC_TYPE_SECURE_BARRIER_CLOSE_ONLY, the device MUST notify the requester by issuing an Application Rejected Request Command. The Application Rejected Request Command is a member of the Application Status Command Class.

4.12.3 Barrier Operator Get Command

This command is used to request the current state of a barrier operator device.

CC:0066.01.02.11.001 The Barrier Operator Report Command MUST be returned in response to this command.

CC:0066.01.02.11.002 This command MUST NOT be issued via multicast addressing.

CC:0066.01.02.11.003 A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_BARRIER_OPERATOR							
Command = BARRIER_OPERATOR_GET							

4.12.4 Barrier Operator Report Command

This command is used to advertise the status of the barrier operator device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_BARRIER_OPERATOR							
Command = BARRIER_OPERATOR_REPORT							
State							

State (8 bits)

CC:0066.01.03.11.001 This field MUST advertise the current state of the device.

CC:0066.01.03.11.002 The encoding of this field MUST be according to Table 18.

Table 18, Barrier Report::State

Value	State	Requirement level	Description	Version
0x00	Closed	REQUIRED	The barrier is in the Closed position	1
0x01..0x63	Stopped at exact Position	RECOMMENDED	0x01 = 1% (Near Closed) 0x63 = 99% (Near Open)	1
0xFC	Closing	REQUIRED	The barrier is closing. The current position is unknown.	1
0xFD	Stopped	REQUIRED	The barrier is stopped. The current position is unknown.	1
0xFE	Opening	REQUIRED	The barrier is opening. The current position is unknown.	1
0xFF	Open	REQUIRED	The barrier is in the Open position	1

CC:0066.01.03.11.003 All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

4.12.5 Barrier Operator Get Signaling Capabilities Supported Command

This command is used to query a device for available subsystems which may be controlled via Z-Wave.

CC:0066.01.04.11.001 The Barrier Operator Report Signaling Capabilites Supported Command MUST be returned in response to this command.

CC:0066.01.04.11.002 This command MUST NOT be issued via multicast addressing.

CC:0066.01.04.11.003 A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_BARRIER_OPERATOR							
Command = BARRIER_OPERATOR_SIGNAL_SUPPORTED_GET							

4.12.6 Barrier Operator Report Signaling Capabilities Supported Command

This command returns a bit mask of signaling subsystem(s) supported by the sending node.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_BARRIER_OPERATOR							
Command = BARRIER_OPERATOR_SIGNAL_SUPPORTED_REPORT							
Bit Mask 1							
….							
Bit Mask N							

Bit Mask (N bytes)

This field is used to advertise the event signaling capabilities supported by the sending node.

CC:0066.01.05.11.004 It MUST be treated as a bitmask.

1. Bit 0 in Bit Mask 1 indicates if subsystem type 0x01 is supported
2. Bit 1 in Bit Mask 1 indicates if subsystem type 0x02 is supported
3. …

CC:0066.01.05.11.001 If the subsystem type is supported, the corresponding bit MUST be set to 1.
If the subsystem type is not supported, the corresponding bit MUST be set to 0.

For the complete list of subsystem types, refer to Table 19.

CC:0066.01.05.11.002 The length of this field depends on the number of subsystems supported by the sending node. The length of this field MUST be in the range 1..32 bytes.

CC:0066.01.05.11.003 A sending node MUST limit the length of this field to represent only those subsystems supported and MUST NOT extend the length of the Bit Mask to pad the packet with 0x00.

4.12.7 Barrier Operator Event Signal Set Command

This command is used to turn on or off an event signaling subsystem that is supported by the device.

CC:0066.01.06.13.001 A supporting node MAY ignore this command if the requested operation violates safety regulartions.

CC:0066.01.06.11.004 UL requirements MUST take precedence over any Set command sent to a subsystem. It is up to the discretion of the device to accept or reject the requested action. If the requested action is rejected, the

CC:0066.01.06.11.001 node MUST notify the requester via the Notification Command Class Report with the Notification {Access Control::Barrier unable to perform requested operation due to UL requirements}.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_BARRIER_OPERATOR							
Command = BARRIER_OPERATOR_SIGNAL_SET							
Subsystem Type							
Subsystem State							

Subsystem Type (8 bits)

CC:0066.01.06.11.002 This field is used to indicate which subsystem type MUST be set at the receiving node. This field MUST comply with Table 19.

Table 19, Barrier Operator Event signal::Subsystem types encoding

Subsystem Type	Description	Version
0x00	NOT SUPPORTED - reserved	1
0x01	The Barrier Device has an Audible Notification subsystem controllable via Z-Wave (For example: Siren)	1
0x02	The Barrier Device has an Visual Notification subsystem controllable via Z-Wave (For example: Flashing Light)	1
0x03.. 0xFF	reserved	1

Subsystem State (8 bits)

CC:0066.01.06.11.003 This field is used to indicate the state that the specified subsystem MUST assume at the receiving node. This field MUST comply with Table 20.

Table 20, Barrier Operator Event Signal::Subsystem states encoding

Subsystem State	Description	Version
0x00	Subsystem OFF	1
…	reserved	1
0xFF	Subsystem ON	1

4.12.8 Barrier Operator Event Signaling Get Command

This command is used to request the state of a signaling subsystem to a supporting node.

CC:0066.01.07.11.001 The Barrier Operator Event Signaling Report Command MUST be returned in response to this command if the specified type is supported.

CC:0066.01.07.11.002 This command MUST NOT be issued via multicast addressing.

CC:0066.01.07.11.003 A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_BARRIER_OPERATOR							
Command = BARRIER_OPERATOR_SIGNAL_GET							
Subsystem Type							

For fields' description, refer to 4.12.7 Barrier Operator Event Signal Set Command.

4.12.9 Barrier Operator Event Signaling Report

This command is used to indicate the state of a notification subsystem of a Barrier Device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_BARRIER_OPERATOR							
Command = BARRIER_OPERATOR_SIGNAL_REPORT							
Subsystem Type							
Subsystem State							

For fields' description, refer to 4.12.7 Barrier Operator Event Signal Set Command

4.13 Basic Command Class, version 1

The Basic Command Class allows a controlling device to operate the primary functionality of a supporting device without any further knowledge.

The Basic Command Class ensures a basic interoperability if no other command class is shared by two devices.

The Basic Command Class is an actuator control command class. Refer to 3.6.

4.13.1 Compatibility considerations

A specific device may not be able to support all Basic CC commands or parameter levels. For instance, a relay can only open fully in response to any non-zero value.

CC:0020.01.00.22.001 Any device SHOULD support the Basic Command Class.

CC:0020.01.00.21.001 A device MUST implement mappings from the Basic Command Class to specific commands according to the advertised generic and specific device class of the Node Info frame. Mappings defined by a Specific Device Class have precedence over the Generic Device Class.

CC:0020.01.00.21.002 For Z-Wave Plus devices, the Basic Command Class MUST be mapped according to the actual Z-Wave Plus Device Type. For further information, refer to [10], [22].

The following sections only present frame formats. For details on the mapping to other command classes, refer to [10] and [22].

4.13.1.1 Node Information Frame (NIF)

CC:0020.01.00.21.003 The Basic Command Class MUST NOT be advertised in the Node Information Frame.
CC:0020.01.00.21.004 The Basic Command Class MUST NOT be advertised in the Security Commands Supported Report (S0 as well as S2)

CC:0020.01.00.21.005 A securely included node MAY support the Basic Command Class at the highest security level but it MUST NOT support the Basic Command Class at any lower security level or non-securely.

4.13.2 Basic Set Command

This command is used to set a value in a supporting device.

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

Value (8 bits)

CC:0020.01.01.12.001 A supporting device SHOULD support all parameter values in the range {0x00..0x63, 0xFF}.
CC:0020.01.01.11.001 A controlling device MUST NOT assume that a receiving device reacts to this command.

CC:0020.01.01.11.002 A receiving device MUST interpret Basic Set parameter values according to the requirements of the Device Class implemented by the device. Refer to [10] and [22].

4.13.3 Basic Get Command

This command is used to request the status of a supporting device.

CC:0020.01.02.11.001 The Basic Report Command MUST be returned in response to this command.

CC:0020.01.02.11.002 This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

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

4.13.4 Basic Report Command

This command is used to advertise the status of the primary functionality of the device.

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

Value (8 bits)

CC:0020.01.03.12.001 The Value field SHOULD advertise the current value of the device hardware; also while in transition to a new target value.

For details on the mapping of values, refer to [10] and [22].

CC:0020.01.03.11.001 A controlling device MUST NOT assume that the Value is identical to a value previously specified in a Set command; not even when a transition has ended.

CC:0020.01.03.11.002 A receiving device MUST interpret the Value field according to Table 21.

Table 21, Basic Report :: Value

Value		Level	State
0	(0x00)	0%	Off
1..99	(0x01..0x63)	1..100%	On
…		Reserved	Reserved
254	(0xFE)	Unknown	Unknown
255	(0xFF)	100%	On

CC:0020.01.03.11.003 All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

4.14 Basic Command Class, version 2

The Basic Command Class allows a controlling device to operate the primary functionality of another device without any further knowledge.

The Basic Command Class ensures a basic interoperability if no other Command Class is shared by two devices.

The Basic Command Class is an actuator control command class. Refer to 3.6.

4.14.1 Compatibility considerations

CC:0020.02.00.21.001 A device supporting Basic Command Class, version 2 MUST support Basic Command Class, version 1. Commands not described in this version remain unchanged from version 1.

Version 2 adds the distinction between the Current Value and the Target State of the device.

CC:0020.02.00.21.002 Version 2 elevates the requirement level so that the Current Value field of the Basic Report command MUST advertise the current value of the device hardware; also while in transition to a new target value.

The compatibility considerations from version 1 also apply to this version. Refer to 4.13.1.

4.14.2 Basic Report Command

This command is used to advertise the status of the primary functionality of the device.

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

Current Value (8 bits)

CC:0020.02.03.11.001 The Current Value field MUST advertise the current value of the device hardware; also while in transition to a new target value.

CC:0020.02.03.11.002 The advertised values may vary depending on the device class implemented by the supporting device. A controlling device MUST interpret the value according to Table 21.For details on the mapping of values, refer to [10] and [22].

CC:0020.02.03.12.001 The Current Value SHOULD be identical to the Target Value when a transition has ended.

Target Value (8 bits)

CC:0020.02.03.11.003 The Target Value field MUST advertise the target value of an ongoing transition or the most recent transition.

CC:0020.02.03.11.004 The advertised values may vary depending on the device class implemented by the supporting device. A controlling device MUST interpret the value according to Table 21.

For details on the mapping of values, refer [10] and [22].

CC:0020.02.03.11.005
CC:0020.02.03.13.001
If queried after receiving a Set command, the Target Value field MUST advertise the target value specified in the Set command. The Target Value MAY change at a later time due to local control or a “Stop” motion control command.

CC:0020.02.03.11.006
If the device is in a motion controlled transition, the Target Value field MUST advertise the min or max value (depending on the direction) according to the command class mappings defined in [10] and [22].

Duration (8 bits)

CC:0020.02.03.12.002
The Duration field SHOULD advertise the time needed to reach the Target Value at the actual transition rate. The encoding of the Duration field MUST be according to Table 8.

4.15 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.

4.15.1 Basic Tariff Information Get Command

This command is used to request current tariff information from the meter.

The Basic Tariff Information Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

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

4.15.2 Basic Tariff Information Report Command

This 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 MUST be ignored by receiving devices.

Reserved

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

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 MUST 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 MUST 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 MUST 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 MUST 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 MUST 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.

4.16 Basic Window Covering Command Class, version 1 [OBSOLETED]

THIS COMMAND CLASS HAS BEEN OBSOLETED

New implementations MUST NOT support this command class.

New window covering device implementations SHOULD support the Window Covering Command Class.

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

4.16.1 Basic Window Covering Start Level Change Command

This command is 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

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

4.16.2 Basic Window Covering Stop Level Change Command

This command is 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							

4.17 Binary Sensor Command Class, version 1 [DEPRECATED]

THIS COMMAND CLASS HAS BEEN DEPRECATED

A device MAY implement this command class, but it is RECOMMENDED that new implementations use the Notification Command Class.
If implementing this command class, it is RECOMMENDED that the Notification Command Class is also implemented.

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

4.17.1 Binary Sensor Get Command

This command is used to request the status of a sensor.

The Binary Sensor Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

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

This command is used to advertise a sensor value.

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.

4.18 Binary Sensor Command Class, version 2 [DEPRECATED]

THIS COMMAND CLASS HAS BEEN DEPRECATED

A device MAY implement this command class, but it is RECOMMENDED that new implementations use the Notification Command Class.
If implementing this command class, it is RECOMMENDED that the Notification Command Class is also implemented.

The Binary Sensor Command Class is 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

This command is used to request the status of the specific sensor device.

The Binary Sensor Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

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

Sensor Type (8 bits)

Sensor Type specifies what type of sensor this command originates from. Refer to Table 22. 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.

Table 22, Binary Sensor Get::Sensor Type encoding

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

4.18.2 Binary Sensor Report Command

This command is used to advertise a sensor value.

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.

4.18.3 Binary Sensor Get Supported Sensor Command

This 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 MUST 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 (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 Sensor Type and must therefore be set to zero.
- Bit 1 in Bit Mask 1 indicates if Sensor Type = 1 (General Purpose) is supported.
- Bit 2 in Bit Mask 1 indicates if Sensor Type = 2 (Smoke) is supported.
- Bit 3 in Bit Mask 1 indicates if Sensor Type = 3 (CO) is supported
- …

If the Sensor Type is supported the bit MUST be set to 1. If the Sensor Type is not supported the bit MUST be set to 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 MUST be determined from the length field in the frame.

4.19 Binary Switch Command Class, version 1

The Binary Switch Command Class is used to control the On/Off state of supporting nodes.

The Binary Switch Command Class is an actuator control Command Class. Refer to 3.6.

1.1.1 Binary Switch Set Command

This command is used to set the On/Off state at the receiving node.

CC:0025.01.01.13.001 A receiving node MAY apply a non-zero duration to the transition from one value to a new value.

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

Value (8 bits)

CC:0025.01.01.11.001 This field is used to specify the On/Off state that the receiving node MUST assume. This field MUST be encoded according to Table 23.

Table 23, Binary Switch Set :: Value

Value		Level	State
0	(0x00)	0%	Off
1..99	(0x01..0x63)	100%	On
255	(0xFF)	100%	On

CC:0025.01.01.11.002 All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

CC:0025.01.01.11.003 If the supporting node uses this Command Class, the values MUST be interpreted as follow:

- The value 0 MUST represent the lowest possible throughput (light, water, sound, etc.)
- The values 1..99 and 255 MUST represent the highest possible throughput (light, water, sound, etc.)

The above value mapping of the Binary Switch Command Class Value allows a controlling node to control a mixed group of Binary Switch and Multilevel Switch supporting nodes via Basic Set commands. Nodes supporting the Binary Switch Command Class turn On or Off while nodes supporting the Multilevel Switch Command Class are set at the specified value level.

1.1.2 Binary Switch Get Command

This command is used to request the current On/Off state from a node.

CC:0025.01.02.11.001 The Binary Switch Report Command MUST be returned in response to this command.

CC:0025.01.02.11.002 This command MUST NOT be issued via multicast addressing.

CC:0025.01.02.11.003 A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

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

1.1.3 Binary Switch Report Command

This command is used to advertise the current On/Off state at the sending node.

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

Value (8 bits)

CC:0025.01.03.11.001 This field is used to advertise the current On/Off state at the sending node. This field MUST be encoded according to Table 24.

Table 24, Binary Switch Report :: Value

Value	Level	State
0 (0x00)	0%	Off
254 (0xFE)	Unknown	Unknown
255 (0xFF)	100%	On

CC:0025.01.03.11.002 All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

CC:0025.01.03.12.001 This field SHOULD advertise the current value of the node's hardware; also while in transition to a new value.

CC:0025.01.03.11.003 A controlling node MUST NOT assume that this field's value is identical to the value previously issued with a Binary Switch Set Command.

4.20 Binary Switch Command Class, version 2

The Binary Switch Command Class is used to control the On/Off state of supporting nodes.

The Binary Switch Command Class is an actuator control Command Class. Refer to 3.6.

4.20.1 Compatibility considerations

CC:0025.02.00.21.001 A node supporting Binary Switch Command Class, version 2 MUST support the Binary Switch Command Class, version 1.

CC:0025.02.00.21.002 Commands not described in this version MUST remain unchanged from version 1.

Version 2 adds duration and target value control and reporting.

4.20.2 Binary Switch Set Command

This command is used to set the binary state at the receiving node.

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

Target Value (8 bits)

CC:0025.02.01.11.001 This field is used to specify the On/Off state that the receiving node MUST assume. This field MUST be encoded according to Table 23.

Duration (8 bits)

CC:0025.02.01.12.001 This field is used to specify the duration that the transition from the current value to the Target Value SHOULD take.

CC:0025.02.01.12.002 A supporting node SHOULD respect the specified duration value.

CC:0025.02.01.11.002 The encoding of the Duration value MUST be according to Table 7.

4.20.3 Binary Switch Report Command

This command is used to advertise the current On/Off state at the sending node.

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

Current Value (8 bits)

CC:0025.02.03.11.001 This field is used to advertise the current On/Off state at the sending node. This field MUST be encoded according to Table 24.

CC:0025.02.03.11.002 The advertised Current Value MUST NOT be updated to the Target Value before the hardware actuator has actually reached the Target Value.

Target Value (8 bits)

CC:0025.02.03.11.003 This field MUST advertise the target value of an ongoing transition or the most recent transition.

CC:0025.02.03.11.004 This field MUST be encoded according to Table 24.

Duration (8 bits)

CC:0025.02.03.12.001 The duration field SHOULD advertise the duration of a transition from the Current Value to the Target Value.

CC:0025.02.03.11.005 The encoding of the Duration field MUST be according to Table 8.

4.21 Binary Toggle Switch Command Class, version 1 [OBSOLETED]

THIS COMMAND CLASS HAS BEEN OBSOLETED

A node MUST NOT implement this Command Class.
New implementations MUST use the Binary Switch Command Class.

The Binary Toggle Switch Command Class is used for toggle-style control of binary actuator devices.

4.21.1 Binary Toggle Switch Set Command

This command is 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							

4.21.2 Binary Toggle Switch Get Command

This command is used to request the state of the load controlled by the device.

The Binary Toggle Switch Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

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

4.21.3 Binary Toggle Switch Report Command

This command is used to advertise the value of a toggle switch.

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 MAY be 0x00 (off) or 0xFF (on).

4.22 Central Scene Command Class, version 1 [OBSOLETED]

THIS COMMAND CLASS HAS BEEN OBSOLETED

CC:005B.01.00.11.001 New implementations MUST support the Central Scene Command Class, version 3 or newer.

The Central Scene Command Class is 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 8.

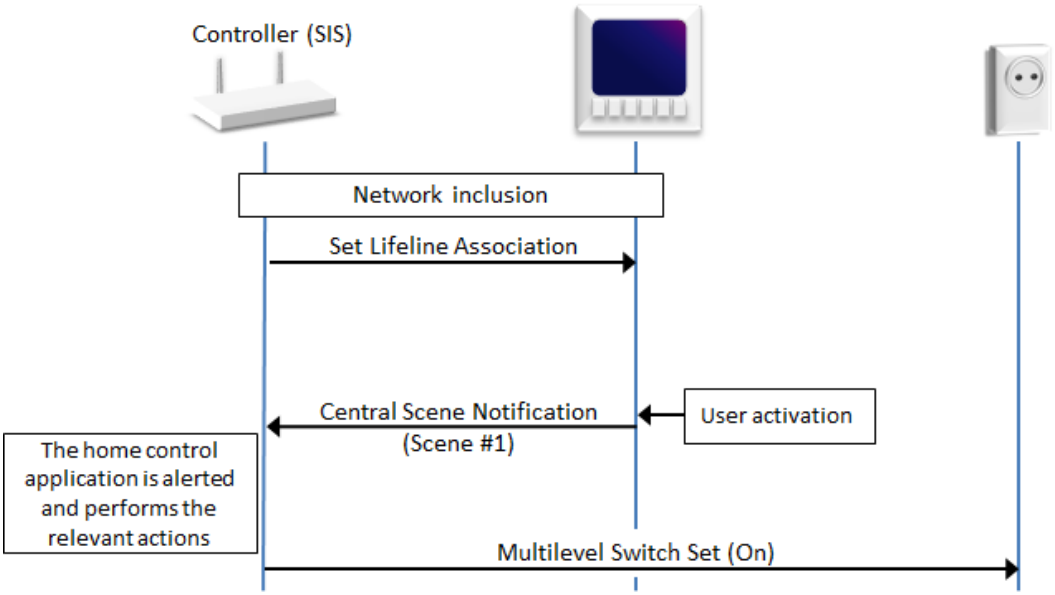


Figure 8, Centralized feedback and control

CC:005B.01.00.13.001 Multiple sending nodes MAY advertise the same Scene number.

CC:005B.01.00.11.002 Therefore, a receiving node MUST interpret the advertised Scene number in combination with the source NodeID of the sending node.

For instance, {Node A, Scene 1} may trigger changes to another scene than {Node B, Scene 1} in the receiving node

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” is increased because the central scene controller may 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.

4.22.1 Central Scene Supported Get Command

This command is used to request the maximum number of scenes that this device supports.

CC:005B.01.01.11.001 The Central Scene Supported Report Command MUST be returned in response to this command.

CC:005B.01.01.11.002 This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

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

This 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 Scenes (8 bits)

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

CC:005B.01.02.11.001 This field MUST advertise the number of available scenes.
Scenes MUST be numbered in the range 1..[Supported Scenes].

CC:005B.01.02.13.001 A device MAY implement virtual buttons via two-button presses, touch screen swipes or other means.
CC:005B.01.02.11.002 Virtual buttons MUST be numbered after physical push buttons or graphical button objects.

CC:005B.01.02.11.003 The advertised number of Supported Scenes MUST cover physical push buttons, graphical button
CC:005B.01.02.13.002 objects as well as virtual buttons. Therefore, the advertised number of Supported Scenes MAY be larger than the number of physical push buttons or graphical button objects.

4.22.3 Central Scene Notification Command

This command 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, the 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.

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 bits)

CC:005B.01.03.11.001 The sequence number MUST be incremented each time a Central Scene Notification Command is issued. The receiving device uses the sequence number to ignore duplicates.

Key Attributes (3 bits)

CC:005B.01.03.11.002 The key Attributes field specifies the state of the key. The field MUST be encoded according to Table 25

Table 25, Central Scene Notification :: Key Attributes

Key Attribute	Description
0x00	Key Pressed
0x01	Key Released
0x02	Key Held Down

CC:005B.01.03.11.003 All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

CC:005B.01.03.11.004 Notifications carrying these Key Attributes MUST comply with Figure 9.

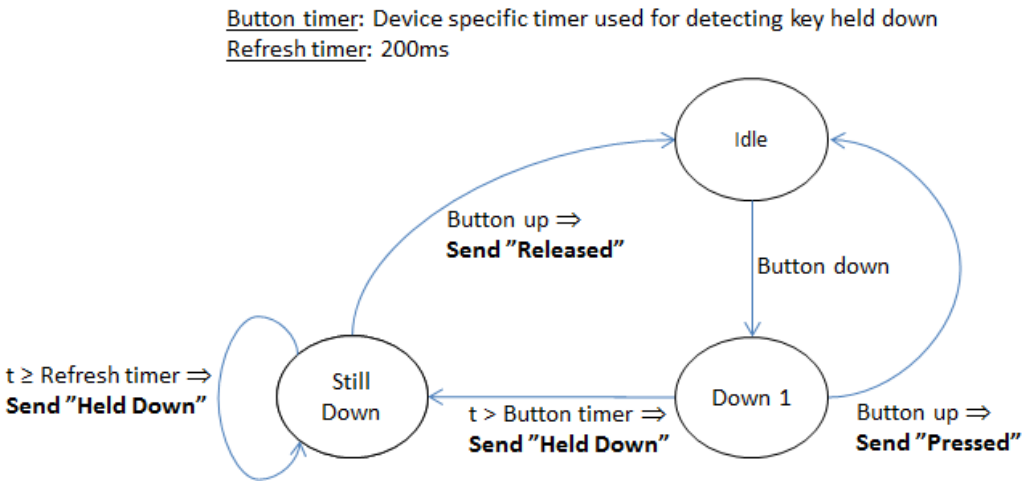


Figure 9, Central Scene version 1 button press decoding and timer management

Reserved

CC:005B.01.03.11.005 This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Scene Number (8 bits)

CC:005B.01.03.11.006 The advertised number of Supported Scenes MUST cover physical push buttons, graphical button objects as well as virtual buttons. Therefore, the advertised number of Supported Scenes MAY be larger than the number of physical push buttons or graphical button objects.
CC:005B.01.03.13.001

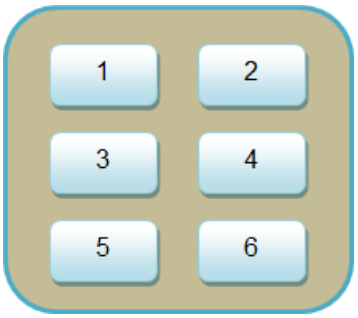


Figure 10, Scene number mapping to button layout

4.23 Central Scene Command Class, version 2 [OBSOLETED]

THIS COMMAND CLASS HAS BEEN OBSOLETED

CC:005B.02.00.11.001 New implementations MUST support the Central Scene Command Class, version 3 or newer.

The Central Scene Command Class is used to communicate central scene activations to a central controller using the lifeline concept.

4.23.1 Compatibility considerations

Central Scene Command Class version 2 is extended on the following areas:

- The Central Scene Supported Report Command is updated to advertise Key Attributes support for each scene.
- Additional Key Attributes are defined.

Commands and paragraphs not mentioned in this version remain unchanged from version 1.

4.23.2 Central Scene Supported Report Command

This command is used to report the maximum number of supported scenes and the Key Attributes supported for each scene.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_CENTRAL_SCENE							
Command = CENTRAL_SCENE_SUPPORTED_REPORT							
Supported Scenes							
Reserved					Number of Bit Mask Bytes		Identical
Supported Key Attributes for Scene 1 – Byte 1							
…							
Supported Key Attributes for Scene 1 – Byte N							
…							
Supported Key Attributes for Scene M – Byte 1							
…							
Supported Key Attributes for Scene M – Byte N							

Fields not described below remain unchanged from version 1.

Reserved

CC:005B.02.02.11.001 This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Number of Bit Mask Bytes (2 bits)

CC:005B.02.02.11.002 This field advertises the size of each “Supported Key Attributes” field measured in bytes. The value MUST be in the range 1..3.

Identical (1 bit)

This field indicates if all scenes are supporting the same Key Attributes:

CC:005B.02.02.11.003 The value 1 MUST indicate that all scenes support the same set of Key Attributes. In this case, the field “Supported Key Attributes for Scene 1” MUST advertise the supported Key Attributes for all scenes

CC:005B.02.02.11.004 The value 0 MUST indicate that scenes support different Key Attributes. In this case, Supported Key Attributes MUST be advertised for each individual scene.

Supported Key Attributes (N bytes)

CC:005B.02.02.11.005 This Bit Mask field advertises the attributes supported by the corresponding scene. The field MUST be encoded according to Table 26.

Table 26: Central Scene Supported Report::Supported Key Attributes

Byte:Bit	Supported Key Attributes
1:0	Key Pressed 1 time
1:1	Key Released.
1:2	Key Held Down.
1:3	Key Pressed 2 times
1:4	Key Pressed 3 times
1:5	Key Pressed 4 times
1:6	Key Pressed 5 times
1:7	Reserved

CC:005B.02.02.11.006 If the Key Released attribute is supported, the Key Held Down attribute MUST also be supported. If the Key Held Down attribute is supported, the Key Released attribute MUST also be supported.

4.23.3 Central Scene Notification Command

This command is used to advertise one or more key events.

CC:005B.02.03.11.001 If the device implements Z-Wave Plus Lifeline support is implemented, the device MUST send the Central Scene Notification Command to the actual Lifeline targets.

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							

Fields not described below remain unchanged from version 1.

Key Attributes (3 bits)

CC:005B.02.03.11.002 This field advertises one or more events detected by the key. The field MUST be encoded according to Table 27

Table 27: Central Scene Notification::Key Attributes

Key Attribute	Description	Version
0x00	Key Pressed 1 time	1
0x01	Key Released	1
0x02	Key Held Down	1
0x03	Key Pressed 2 times	2
0x04	Key Pressed 3 times	2
0x05	Key Pressed 4 times	2
0x06	Key Pressed 5 times	2

CC:005B.02.03.11.003 All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

CC:005B.02.03.11.004 Notification carrying these Key Attributes MUST comply with Figure 11.

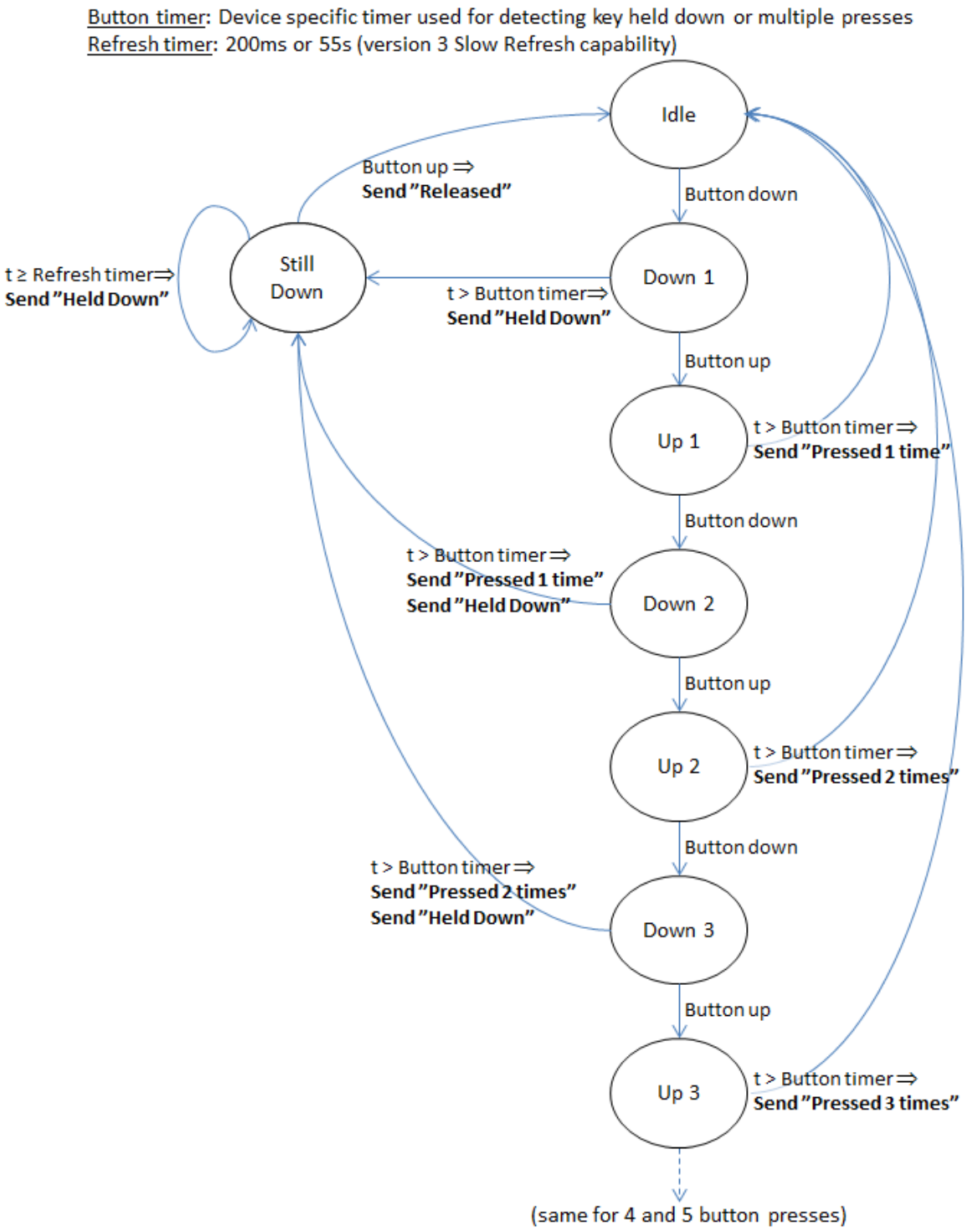


Figure 11, Central Scene button press decoding and timer management

4.24 Central Scene Command Class, version 3

The Central Scene Command Class, version 3 extends version 2 by adding a new mode for the transmission of Key Held Down notifications. This allows a node to send Key Held Down refreshes at a low rate for the duration of the key held down.

The purpose of this new feature is to provide improved compatibility with FLiRS beaming and network routing.

Central Scene Command Class, version 3 introduces three new commands to allow a controller to query and configure node capabilities.

- Central Scene Configuration Set Command
- Central Scene Configuration Get Command
- Central Scene Configuration Report Command

Central Scene Command Class, version 3 extends the following commands for a scene launching node to advertise its optional capabilities.

- Central Scene Supported Report Command
- Central Scene Notification Command

4.24.1 Compatibility considerations

CC:005B.03.00.22.001 A scene controller SHOULD NOT try to detect multiple key presses from the reception of successive key press notifications in any single or multiple key attribute combination. FLiRS beaming, collisions or routing may add delays between messages and render the timing between notifications unreliable.

Version 1 and version 2 required to send Key Held Down notifications commands every 200ms. It has been found that this requirement cannot be observed when the network is using FLiRS beaming or routing. Further, issuing the 200ms refreshes also degrades general network reliability and availability.

Version 3 introduces the Slow Refresh Capability, which allows a node to send refreshes every 55 seconds instead of 200ms. For details about the Slow Refresh capability, refer to 4.24.7.

CC:005B.03.00.21.001 In order to ensure a functioning network, the Slow Refresh capability MUST be enabled by default after inclusion.

CC:005B.03.00.22.002 When creating associations from a version 3 node, the node creating the association SHOULD enable the Slow Refresh capability in the node if the association destination does also support Central Scene Command Class, version 3. When using the version 3 Slow Refresh capability, the controller relies on the reception of a Key Up notification instead of the Key Held Down refreshes. The new mode still sends refreshes at a low rate, for the controller to detect if a node failed after sending a Held Down Key notification. This means that a version 2 controlling node may time out when receiving Key Held Down Notifications from a version 3 node.

However, it has been found that there exist nodes supporting version 1 or version 2 which do not send Central Scene Notification refreshes every 200ms when the Key Held Down notification is issued.

CC:005B.03.00.22.003 A controller SHOULD apply an adaptive approach based on the reception of the Key Released Notification. Initially, the controller SHOULD time out if not receiving any Key Held Down Notification refresh after 400ms and consider this to be a Key Up Notification. If, however, the controller subsequently receives a Key Released Notification, the controller SHOULD consider the sending node to be operating with the Slow Refresh capability enabled.

4.24.1.1 Central Scene Configuration commands support

If a node supports Slow Refresh Capability (that allows a node to send refresh every 55 seconds instead of 200 ms), may not allow users to turn off the 'Slow Refresh' functionality. If this is the case, the node MAY choose to ignore the Central Scene Configuration Set Command with 'Slow Refresh' bit set to zero.

If the node chooses to ignore the Central Scene Configuration Set Command, the node MUST return an application Reject Request Command when receiving Central Scene Configuration Set command (if it received without Supervision encapsulation).

4.24.1.2 Multi Channel considerations

CC:005B.03.00.22.004 Multi Channel End Points SHOULD NOT support the Central Scene Command Class.

4.24.2 Central Scene Configuration Set Command

This command is used to configure the use of optional node capabilities for scene notifications.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_CENTRAL_SCENE							
Command = CENTRAL_SCENE_CONFIGURATION_SET							
Slow Refresh	Reserved						

Slow Refresh (1 bit)

This flag is used to configure the use of the Slow Refresh capability.

CC:005B.03.04.11.001 The value 1 MUST indicate that the scene launching node MUST use Slow Refresh.

The value 0 MUST indicate that the scene launching node MUST NOT use Slow Refresh.

Reserved

CC:005B.03.04.11.002 This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

4.24.3 Central Scene Configuration Get Command

This command is used to query the configuration of optional node capabilities for scene notifications.

CC:005B.03.05.11.001 The Central Scene Configuration Report Command MUST be returned in response to this command.

CC:005B.03.05.11.002 This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

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

4.24.4 Central Scene Configuration Report Command

This command is used to advertise the configuration of optional node capabilities for scene notifications.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_CENTRAL_SCENE							
Command = CENTRAL_SCENE_CONFIGURATION_REPORT							
Slow Refresh	Reserved						

Slow Refresh (1 bit)

This flag is used to advertise the use of the Slow Refresh capability.

CC:005B.03.06.11.001 The value 1 MUST indicate that the scene launching node MUST use Slow Refresh.
The value 0 MUST indicate that the scene launching node MUST NOT use Slow Refresh capability.

Reserved

CC:005B.03.06.11.002 This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

4.24.5 Central Scene Supported Get Command

This command is used to request the maximum number of scenes that this device supports.

CC:005B.03.01.11.001 The Central Scene Supported Report Command MUST be returned in response to this command.

CC:005B.03.01.11.002 This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

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

4.24.6 Central Scene Supported Report Command

This command is used to report the maximum number of supported scenes and the Key Attributes supported for each scene.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_CENTRAL_SCENE							
Command = CENTRAL_SCENE_SUPPORTED_REPORT							
Supported Scenes							
Slow Refresh Support	Reserved				Number of Bit Mask Bytes		Identical
Supported Key Attributes for Scene 1 – Byte 1							
…							
Supported Key Attributes for Scene 1 – Byte N							
…							
Supported Key Attributes for Scene M – Byte 1							
…							
Supported Key Attributes for Scene M – Byte N							

All fields not described below are the same as in version 2.

Slow Refresh Support (1 bit)

This field indicates whether the node supports the Slow Refresh capability.

CC:005B.03.02.11.001 The value 1 MUST indicate that the node supports the Slow Refresh capability.
The value 0 MUST indicate that the node does not support the Slow Refresh capability.

4.24.7 Central Scene Notification Command

This command is used to advertise a key event.

CC:005B.03.03.11.001 If the device implements Z-Wave Plus Lifeline support, the device MUST send the Central Scene Notification Command to the actual Lifeline targets.

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

All fields not described below are the same as in version 2.

Slow Refresh (1 bit)

CC:005B.03.03.11.002 CC:005B.03.03.11.003 This flag is used to advertise if the node is sending Key Held Down notifications at a slow rate. A sending node MUST always set this field according to the configured mode.
A receiving node MUST ignore this field if the command is not carrying the Key Held Down key attribute.

If the Slow Refresh field is 0:

CC:005B.03.03.12.002
- A new Key Held Down notification SHOULD be sent every 200ms until the key is released.

CC:005B.03.03.11.005
- The Sequence Number field MUST be updated at each notification transmission.

CC:005B.03.03.12.001
- If not receiving a new Key Held Down notification within 400ms, a controlling node SHOULD use an adaptive timeout approach as described in 4.24.1.

If the Slow Refresh field is 1:

CC:005B.03.03.11.006
- A new Key Held Down notification MUST be sent every 55 seconds until the key is released.

CC:005B.03.03.11.007
- The Sequence Number field MUST be updated at each notification refresh.

CC:005B.03.03.11.008
- If not receiving a new Key Held Down notification within 60 seconds after the most recent Key Held Down notification, a receiving node MUST respond as if it received a Key Release notification.

4.25 Climate Control Schedule Command Class, version 1 [DEPRECATED]

THIS COMMAND CLASS HAS BEEN DEPRECATED

A device MAY implement this command class, but it is RECOMMENDED that new implementations use the Schedule Command Class.
If implementing this command class, it is RECOMMENDED that the Schedule Command Class is also implemented.

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 are exchanged using the Schedule Commands.
Overrides of schedules are exchanged using the Schedule Override Commands.
Detection of updated schedules is 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.

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

4.25.1 Schedule Set Command

This command is 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” MUST be placed last. Duplicates MUST NOT 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 8 Byte 1							
Switchpoint 8 Byte 2							
Switchpoint 8 Byte 3							

Weekday (3 bits)

This field MUST be encoded according to Table 28.

Table 28, Climate Control Schedule Set::Weekday encoding

Value	Description
0x00	Reserved
0x01	Monday
0x02	Tuesday
0x03	Wednesday
0x04	Thursday
0x05	Friday
0x06	Saturday
0x07	Sunday

Switchpoint (9*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)

This field is used to specify the hour of the of the actual switchpoint. This field MUST be in the range 0..23.

Minute (6 bits)

This field is used to specifiy the minute of the actual switchpoint. This field MUST be in the range 0..59.

Schedule State (8 bits)

Schedule State uses the Schedule State type format, see section 4.25. If Schedule State has the value of “Unused”, the Hour and Minute field MUST be ignored. Once a Schedule State of “Unused” is encountered, the parsing of switchpoints must stop.

4.25.2 Schedule Get Command

This command is used to request the climate control schedule in a device for a specific weekday.

The Schedule Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

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

For fields’ description, refer to 4.25.1 Schedule Set Command

4.25.3 Schedule Report Command

This command is 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 8 Byte 1							
Switchpoint 8 Byte 2							
Switchpoint 8 Byte 3							

For fields' description, refer to 4.25.1 Schedule Set Command

4.25.4 Schedule Changed Get Command

This command is used to check if the climate control schedule has changed.

The Schedule Changed Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

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

4.25.5 Schedule Changed Report Command

This command is 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 MUST update its ChangeCounter, and the other devices MUST 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:

Value	Description
0x00	The climate control schedule change mechanism is temporarily disabled by the override function.
0x01..0xFF	The climate control schedule change mechanism is enabled. ChangeCounter is incremented by one every time climate control schedule changes. When ChangeCounter eventually reach 0xFF, the next increment MUST rollover to 0x01.

When a device is fresh and has no climate control schedule it MUST retrieve a climate control schedule using the Schedule Get Command and it MUST 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.

4.25.6 Schedule Override Set Command

This command is 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)

This field MUST be encoded according to Table 29.

Table 29, Climate Control Schedule Override Set::Override Type encoding

Value	Description
0x00	No override
0x01	Temporary override
0x02	Permanent override
0x03	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 MUST be notified in an unsolicited SCHEDULE_OVERRIDE_REPORT as specified in the following sequence diagram:

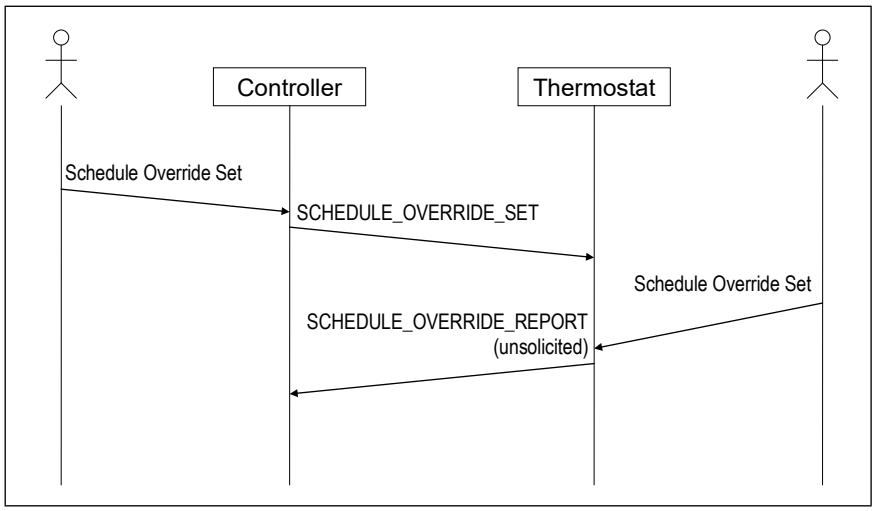


Figure 12, Sequence diagram for cancellation of a Schedule Override Set

Override State (8 bits)

The Override State uses the Schedule State type format, see section 4.25

4.25.7 Schedule Override Get Command

This command is used to request the override, currently in use in a device.

The Schedule Override Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

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

4.25.8 Schedule Override Report Command

The Schedule Override Report Command is used to report 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_REPORT							
Reserved						Override Type	
Override State							

Override Type (2 bits)

This field MUST be encoded according to Table 29.

Both temporary and permanent overrides MAY be cancelled in the device, which receives the SCHEDULE_OVERRIDE_SET. This cancellation MUST be notified in an unsolicited SCHEDULE_OVERRIDE_REPORT as shown on figure in section 4.25.6

Override State (8 bits)

The Override State uses the Schedule State type format, see section 4.25

4.26 Clock Command Class, version 1

The Clock Command Class is used to implement a simple clock functionality.

1.1.1 Interoperability considerations

CC:0081.01.00.32.001 A controlling node SHOULD configure the current time and weekday in a supporting node during node commissioning.

It has been found that some version 1 nodes issue Clock Get Command to controllers in order to learn the current time and use the returned Clock Report Command as a Clock Set Command.

CC:0081.01.00.32.002 A supporting node SHOULD NOT learn the current time with a Clock Get Command.

CC:0081.01.00.32.003 A node SHOULD control the Time Command Class to read the current time from supporting nodes (time servers) or support the Clock Command Class and wait to be set the current time by a controlling node.

CC:0081.01.00.32.004 A Z-Wave Plus node SHOULD issue a Clock Report Command via the Lifeline Association Group if they suspect to have inaccurate time and/or weekdays (e.g. after battery removal).

CC:0081.01.00.32.005 A controlling node SHOULD compare the received time and weekday with its current time and set the time again at the supporting node if a deviation is observed (e.g. different weekday or more than a minute difference)

4.26.1 Multi Channel Considerations

CC:0081.01.00.12.001 Multi Channel End Points SHOULD NOT support the Clock Command Class.

4.26.2 Clock Set Command

This command is used to set the current time in a supporting node.

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

Weekday (3 bits)

CC:0081.01.04.11.001 This field is used to indicate the current weekday that the receiving node MUST assume.

CC:0081.01.04.11.002 This field MUST comply with Table 30

Table 30, Clock Set Command::Weekday encoding

Value	Description
0x00	Unused/unknown This value may be specified if the weekday is not used or not known
0x01	Monday
0x02	Tuesday
0x03	Wednesday
0x04	Thursday
0x05	Friday
0x06	Saturday
0x07	Sunday

Hour (5 bits)

CC:0081.01.04.11.003 This field is used to indicate the hour of the current time that the receiving node MUST assume.

CC:0081.01.04.11.004 This field MUST be in the range 0..23.

Minute (8 bits)

CC:0081.01.04.11.005 This field is used to indicate the minute of the current time that the receiving node MUST assume.

CC:0081.01.04.11.006 This field MUST be in the range 0..59.

CC:0081.01.04.12.001 A sending node knowing the current time with seconds precision SHOULD round its current time to the nearest minute when sending this command.

1.1.2 Clock Get Command

This command is used to request the current time set at a supporting node.

CC:0081.01.05.11.001 The Clock Report Command MUST be returned in response to this command.

CC:0081.01.05.11.002 This command MUST NOT be issued via multicast addressing.

CC:0081.01.05.11.003 A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

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

1.1.3 Clock Report Command

This command is used to advertise the current time set at the sending node.

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

For fields' description, refer to Clock Set Command

CC:0081.01.06.11.001 A sending node MUST comply with fields' description from Clock Set Command

4.27 Color Switch Command Class, version 1

The Color Switch Command Class is used to control color capable devices.

The Color Switch Command Class manipulates the color components of a device. Each color component is scaled by the brightness level previously set by a Multilevel Switch Set, Binary Switch Set or Basic Set Command.

The Color Switch Command Class is an actuator control command class. Refer to 3.6.

4.27.1 Compatibility considerations

The Color Switch Command Class, version 1 was previously named the Color Control Command Class, version 1. The Color Switch Command Class, version 1 is binary compatible with the Color Control Command Class, version 1.

4.27.1.1 Command Class dependencies

CC:0033.01.00.21.001 Nodes supporting the Color Switch Command Class, version 1 MUST support one of the following command classes in order to control the color brightness:

- Multilevel Switch Command Class, version 2
- Binary Switch Command Cass, version 2.

4.27.2 Interoperability considerations

CC:0033.01.00.31.001 The Color Switch Command Class MUST be treated as a separate Command Class.

CC:0033.01.00.31.002 Basic and Binary/Multilevel Switch Command Class commands MUST NOT affect color component levels controlled by the Color Switch Command Class.

CC:0033.01.00.31.003 The Color Switch Command Class commands MUST NOT affect the brightness level controlled by Basic or Binary/Multilevel Switch Command Classes.

CC:0033.01.00.31.004 If the brightness level is requested via a Basic Get or Binary/Multilevel Switch Get command, the reported value MUST reflect the brightness level previously set via Basic or Multilevel Switch commands.

CC:0033.01.00.31.005 If a color component is requested via the Color Switch Get command, the reported value MUST reflect the color component level previously set via Color Switch commands.

CC:0033.01.00.31.006 An implementation MUST scale color component levels by the brightness level . Thus, to achieve a resulting light level of 100% for a given color component, the color component level must be set to 100% (via the Color Switch Command Class) and the brightness level must be set to 100% (via Basic or Multilevel Switch Command Class).

CC:0033.01.00.32.001 A controlling device SHOULD specify the highest possible color component levels of a given color tone to achieve the highest light yield when scaled by the brightness level.

CC:0033.01.00.33.001
CC:0033.01.00.31.007 The controlling device MAY however limit color component levels to achieve the same brightness for blended color tones as for pure colors. For this reason, a supporting device MUST NOT normalize color components manipulated by the Color Switch Command Class.
Likewise, a supporting device implementing indexed colors should reduce its internal color component levels for blended colors to achieve the same brightness for all color tones.

4.27.3 Color Switch Supported Get Command

This command is used to request the supported color components of a device.

CC:0033.01.01.11.001 The Color Switch Supported Report command MUST be returned in response to this command.

CC:0033.01.01.11.002 This command MUST NOT be issued via multicast addressing.

CC:0033.01.01.11.003 A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_COLOR							
Command = SWITCH_COLOR_SUPPORTED_GET							

4.27.4 Color Switch Supported Report Command

This command is used to report the supported color components of a device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_COLOR							
Command = SWITCH_COLOR_SUPPORTED_REPORT							
Color Component Mask 1							
Color Component Mask 2							

Color Component Mask (variable length)

CC:0033.01.02.11.001 The Color Component Mask field MUST advertise the color components supported by the device.

- Bit 0 in Bit Mask 1 indicates if color component 0 is supported
- Bit 1 in Bit Mask 1 indicates if color component 1 is supported
- …

For the definition of Color Component IDs, refer to section 4.27.7.

4.27.5 Color Switch Get Command

This command is used to request the status of a specified color component.

CC:0033.01.03.11.001 The Color Switch Report Command MUST be returned in response to this command.

CC:0033.01.03.11.002 This command MUST NOT be issued via multicast addressing.

CC:0033.01.03.11.003 A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_COLOR							
Command = SWITCH_COLOR_GET							
Color Component ID							

Color Component ID (8 bits)

CC:0033.01.03.11.004 This field MUST specify the color component for which the status is requested.

For the definition of Color Component IDs, refer to section 4.27.7.

4.27.6 Color Switch Report Command

This command is used to advertise the status of a color component.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_COLOR							
Command = SWITCH_COLOR_REPORT							
Color Component ID							
Value							

Color Component ID (8 bits)

CC:0033.01.04.11.001 This field MUST advertise the color component covered by this report.

For the definition of Color Component IDs, refer to section 4.27.7.

Value (8 bits)

CC:0033.01.04.11.002 The Value field MUST advertise the value of the color component identified by the Color Component ID.

CC:0033.01.04.12.001 The Value field SHOULD advertise the current value of the device hardware; also while in transition to a new target value.

CC:0033.01.04.11.003 A controlling device MUST NOT assume that the Value is identical to a value previously issued with a Set command; not even when a transition has ended.

4.27.7 Color Switch Set Command

This command is used to control one or more color components in a device.

CC:0033.01.05.11.001 Color component levels MUST be scaled by the brightness level. Refer to 4.27.2.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_COLOR							
Command = SWITCH_COLOR_SET							
Reserved			Color Component Count				
Color Component ID 1							
Value 1							
…							
Color Component ID N							
Value N							

Reserved

CC:0033.01.05.11.002 This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Color Component Count (5 bits)

CC:0033.01.05.11.003 This field MUST specify the number of (Color Component ID, Value) datasets contained in the Color Switch Set command.

Color Component ID (8 bits)

CC:0033.01.05.11.004 This field MUST specify the color component to receive a new value. This field MUST be encoded according to Table 31.

Table 31, Color Switch Component IDs

Component ID	Label	Value range
0	Warm White	0x00..0xFF
1	Cold White	0x00..0xFF
2	Red	0x00..0xFF
3	Green	0x00..0xFF
4	Blue	0x00..0xFF
5	Amber (for 6ch Color mixing)	0x00..0xFF
6	Cyan (for 6ch Color mixing)	0x00..0xFF
7	Purple (for 6ch Color mixing)	0x00..0xFF
8	Indexed Color [OBSOLETED] A supporting node MUST NOT support indexed colors	0x00 – 0xFF: Color Index 0-255

CC:0033.01.05.11.008

CC:0033.01.05.11.005 All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

Value (8 bits)

CC:0033.01.05.11.006 This field MUST specify the value of the color component identified by the Color Component ID field.

CC:0033.01.05.11.007 Unlike other actuator control command classes, the Value used by the Color Switch Command Class spans the entire range from 0x00 to 0xFF. Thus, except for the Indexed Color, the value 0x00 MUST yield to 0% and 0xFF MUST yield to 100% of the actual color component.

4.27.8 Color Switch Start Level Change Command

This command is used to initiate a transition of one color component to a new level.

CC:0033.01.06.11.001 A receiving device MUST initiate the transition to a new value for the specified Color Component ID.

CC:0033.01.06.13.001 The device MAY apply a non-zero duration to the transition from one value to a new value.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_COLOR							
Command = SWITCH_COLOR_START_LEVEL_CHANGE							
Reserved	Up/ Down	Ignore Start Level	Reserved				
Color Component ID							
Start Level							

Up/Down (1 bit)

CC:0033.01.06.11.002 This field MUST specify the direction of the level change.

CC:0033.01.06.11.003 If the Up/Down bit is set to 0 the level change MUST be increasing.

CC:0033.01.06.11.004 If the Up/Down bit is set to 1 the level change MUST be decreasing.

Ignore Start Level (1 bit)

CC:0033.01.06.12.001 A receiving device SHOULD respect the Start Level if the Ignore Start Level bit is 0.

CC:0033.01.06.11.005 A receiving device MUST ignore the Start Level if the Ignore Start Level bit is 1.

CC:0033.01.06.12.002 A controlling device SHOULD set the Ignore Start Level bit to 1.

Reserved

CC:0033.01.06.11.006 This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Color Component ID (8 bits)

CC:0033.01.06.11.007 This field MUST specify the color component to start a transition.

For the definition of Color Component IDs, refer to section 4.27.7.

Start Level (8 bits)

CC:0033.01.06.11.008 The Start Level field MUST specify the initial value of the level change.

4.27.9 Color Switch Stop Level Change Command

This command is used to stop an ongoing transition initiated by a Color Switch Start Level Change Command.

CC:0033.01.07.11.001 A receiving device MUST stop the transition if the specified Color Component ID is currently in transition to a new value.

CC:0033.01.07.11.002 A receiving device MUST NOT stop ongoing transitions for other Color Component IDs than the one specified.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_COLOR							
Command = SWITCH_COLOR_STOP_LEVEL_CHANGE							
Color Component ID							

Color Component ID (8 bits)

CC:0033.01.07.11.003 This field MUST specify the color component to stop a transition.

For the definition of Color Component IDs, refer to section 4.27.7.

4.28 Color Switch Command Class, version 2

The Color Switch Command Class, version 2 is used to control color capable devices.

The Color Switch Command Class is an actuator control command class. Refer to 3.6.

4.28.1 Compatibility considerations

CC:0033.02.00.21.001 A device supporting Color Switch Command Class, version 2 MUST support Color Switch Command Class, version 1.

Version 2 adds a duration parameter to the Color Switch Set Command.

Commands not described in this version remain unchanged from version 1.

4.28.1.1 Command Class dependencies

CC:0033.02.00.21.002 Nodes supporting the Color Switch Command Class, version 2 MUST support one of the following command classes in order to control the color brightness:

- Multilevel Switch Command Class, version 2
- Binary Switch Command Cass, version 2.

4.28.2 Interoperability considerations

Refer to 4.27.2.

4.28.3 Color Switch Set Command

This command is used to control one or more color components in a device.

CC:0033.02.05.11.001 Color component levels MUST be scaled by the brightness level. Refer to 4.27.2.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_COLOR							
Command = SWITCH_COLOR_SET							
Reserved				Color Component Count			
Color Component ID 1							
Value 1							
...							
Color Component ID N							
Value N							
Duration							

Reserved

CC:0033.02.05.11.002 This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Color Component Count (5 bits)

CC:0033.02.05.11.003 This field MUST specify the number of (Color Component ID, Value) datasets contained in the Color Switch Set command.

Color Component ID (8 bits)

CC:0033.02.05.11.004 This field MUST specify the color component to receive a new value.

For the definition of Color Component IDs, refer to section 4.27.7.

Value (8 bits)

CC:0033.02.05.11.005 This field MUST specify the value of the color component identified by the Component ID field.

Refer to section Table 31.

Duration (8 bits)

CC:0033.02.05.11.006 The Duration field MUST specify the time that the transition should take from the current value to the new target value.

CC:0033.02.05.12.001 A supporting device SHOULD respect the specified Duration value.

CC:0033.02.05.11.007 The encoding of the Duration field MUST be according to Table 7.

CC:0033.02.05.12.002 The factory default duration SHOULD be the same as the duration used for the Color Switch Set command, version 1.

4.29 Color Switch Command Class, version 3

The Color Switch Command Class is used to control color capable devices.

The Color Switch Command Class manipulates the color components of a device. Each color component is scaled by the brightness level previously set by a Multilevel Switch Set or Basic Set command.

The Color Switch Command Class is an actuator control command class. Refer to 3.6.

4.29.1 Compatibility considerations

CC:0033.03.00.21.001 A node supporting Color Switch Command Class, version 3 MUST support the Color Switch Command Class, version 2.

Version 3 adds duration and target value reporting to the Color Switch Report Command and adds duration control to the Color Switch Start Level Change Command.

Commands not described in this version remain unchanged from version 2.

CC:0033.03.00.23.001 A node receiving a version 1 Color Switch Set command MAY apply a factory default duration to the transition.

4.29.1.1 Command Class dependencies

CC:0033.03.00.21.002 Nodes supporting the Color Switch Command Class, version 3 MUST support one of the following Command Classes in order to control the color brightness:

- Multilevel Switch Command Class, version 4
- Binary Switch Command Cass, version 2.

4.29.2 Interoperability considerations

Refer to 4.27.2

4.29.3 Color Switch Report Command

This command is used to advertise the status of a color component.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_COLOR							
Command = SWITCH_COLOR_REPORT							
Color Component ID							
Current Value							
Target Value							
Duration							

Color Component ID (8 bits)

CC:0033.03.04.11.001 This field MUST advertise the color component covered by this report.

Current Value (8 bits)

CC:0033.03.04.11.002 The Current Value field MUST advertise the current value of the color component identified by the Color Component ID.

Refer to the Color Switch Set command for valid values.

CC:0033.03.04.12.001 The Current Value SHOULD be identical to the Target Value when a transition has ended.

Target Value (8 bits)

CC:0033.03.04.11.003 The Target Value field MUST advertise the target value of an ongoing transition or the most recent transition for the advertised Color Component ID.

Refer to the Color Switch Set command for valid values.

CC:0033.03.04.11.004 If a transition is initiated in an interactive fashion via a local user interface or via a Start Level Change command, the advertised Target Value MUST be 0x00 or 0xFF, depending on the direction.

CC:0033.03.04.12.002 The Current Value SHOULD be identical to the Target Value when a transition has ended.

Duration (8 bits)

CC:0033.03.04.12.003 CC:0033.03.04.11.005 The Duration field SHOULD advertise the time needed to reach the Target Value at the actual transition rate. The encoding of the Duration field MUST be according to Table 8.

4.29.4 Color Switch Start Level Change Command

This command is used to initiate a transition of one color component to a new value.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_COLOR							
Command = SWITCH_COLOR_START_LEVEL_CHANGE							
Res	Up/ Down	Ignore Start Level	Reserved				
Color Component ID							
Start Level							
Duration							

Up/Down (1 bit)

CC:0033.03.06.11.001 This field MUST specify the direction of the level change.
CC:0033.03.06.11.002 If the Up/Down bit is set to 0 the level change MUST be increasing.
CC:0033.03.06.11.003 If the Up/Down bit is set to 1 the level change MUST be decreasing.

Ignore Start Level (1 bit)

CC:0033.03.06.12.001 A receiving device SHOULD respect the Start Level if the Ignore Start Level bit is 0.

CC:0033.03.06.11.004 A receiving device MUST ignore the Start Level if the Ignore Start Level bit is 1.

CC:0033.03.06.12.002 A controlling device SHOULD set the Ignore Start Level bit to 1.

Res

CC:0033.03.06.11.005 This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Color Component ID (8 bits)

Refer to Color Switch Set command.

Start Level (8 bits)

CC:0033.03.06.11.006 The Start Level field MUST specify the initial value of the level change.

Duration (8 bits)

CC:0033.03.06.11.007 The level change rate MUST be calculated to match a level change from 0x00 to 0xFF during the time specified by the Duration field.

CC:0033.03.06.12.003 A supporting device SHOULD respect the specified Duration value.

For encoding of the Duration value, refer to the Color Switch Set Command.

4.30 Configuration Command Class, version 1

The Configuration Command Class allows product specific configuration parameters to be changed. One example could be the default dimming rate of a light dimmer.

CC:0070.01.00.11.001
CC:0070.01.00.11.002
Configuration parameters MUST be specified in the product documentation. Configuration parameters accessed via this command class MUST NOT replace similar commands provided by other existing Command Classes.

CC:0070.01.00.11.003
A device MUST be able to operate with default factory configuration parameter values.

CC:0070.01.00.12.001
CC:0070.01.00.12.002
It is RECOMMENDED that configuration parameters can be manipulated via a local user interface.
It is RECOMMENDED that default factory configuration parameter values can be restored via a local user interface.

4.30.1 Compatibility considerations

4.30.1.1 "Default" flag

CC:0070.01.00.23.001
Earlier text revisions of the Configuration Command Class, versions 1-3 presented conflicting interpretations of the default field. Controllers should be aware that nodes with version 3 or less MAY reset all configuration parameters to default when receiving a Configuration Set or Configuration Bulk Set Command with the Default bit set to 1.

CC:0070.01.00.22.001
A controller SHOULD probe a node in order to discover what behavior it implements. The following steps are RECOMMENDED:

- If the device is version 1- 2, scan the node for available configuration parameters. Else discover the available parameters thanks to the Configuration Properties Get Command.
- Determine how the default bit works if the node implements 2 or more parameters

Step 1:

- One at a time, issue a Configuration Set for a parameter number, and check whether the value can be read back. Try each parameter for 1, 2 and 4 bytes sizes.
- Store the discovered parameter list for future use.

Step 2:

- Choose 2 parameters and issue a Configuration Set with the default bit set to 1 for each parameter.
- Read back both parameters value with a Configuration Get.
- Try to modify both parameters to non-default values (it is recommended to try the default value ± 1)
- Send a Configuration Set for one of parameters with the default bit set to 1.
- Read back both parameters and check whether one or both returned to the default value

4.30.2 Configuration Set Command

The Configuration Set Command is used to set the value of a configuration parameter.

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 N							

Parameter Number (8 bits)

CC:0070.01.04.13.001
CC:0070.01.04.12.001

This field is used to specify the actual configuration parameter. Parameter Numbers MAY be product specific. Parameter Numbers SHOULD be assigned in a sequence starting from 1.

Default (1 bit)

This field is used to specify if the default value is to be restored for the specified configuration parameter.

CC:0070.01.04.11.001
CC:0070.01.04.11.002

The value 1 MUST indicate that default factory settings must be restored for the specified Parameter Number. In this case, the Configuration Value field MUST be ignored and the receiving node MUST reset the specified parameter and SHOULD NOT reset any other parameters.

CC:0070.01.04.11.003

The value 0 MUST indicate that the specified Parameter Number must assume the value specified by the Configuration Value field.

Reserved

CC:0070.01.04.11.004

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Size (3 bits)

CC:0070.01.04.11.005

This field is used to specify the size of the actual parameter. The advertised size of a given parameter MUST always be the same. The value of this field MUST comply with Table 32.

Table 32, Configuration Set::Size encoding

Size	Size of Value field
1	8 bit
2	16 bit
4	32 bit

CC:0070.01.04.11.006 All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

Configuration Value (N bytes)

CC:0070.01.04.11.007 This field carries the value to be assigned. The size of the field MUST comply with the size advertised by the Size field.

CC:0070.01.04.11.008 The field MUST carry a signed value. The binary encoding of the signed value MUST comply with Table 10. The field Value 1 MUST be the most significant byte.

4.30.3 Configuration Get Command

This command is used to query the value of a configuration parameter.

CC:0070.01.05.11.001 The Configuration Report Command MUST be returned in response to this command.

CC:0070.01.05.11.002 This command MUST NOT be issued via multicast addressing.

CC:0070.01.05.11.003 A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

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

Parameter Number (8 bits)

CC:0070.01.05.13.001
CC:0070.01.05.12.001 This field is used to specify the requested configuration parameter. Parameter Numbers MAY be product specific. Parameter Numbers SHOULD be assigned in a sequence starting from 1.

CC:0070.01.05.12.002 A node receiving this command for an unsupported parameter SHOULD return a Report advertising the value of the first available parameter in the node.

4.30.4 Configuration Report Command

This command is used to advertise the actual value of the advertised parameter.

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 N							

Refer to the Configuration Set Command for parameter details.

4.31 Configuration Command Class, version 2

The Configuration Command Class allows product specific configuration parameters to be changed. One example could be the default dimming rate of a light dimmer.

CC:0070.02.00.11.001
CC:0070.02.00.11.002
Configuration parameters MUST be specified in the product documentation. Configuration parameters accessed via this command class MUST NOT replace similar commands provided by other existing Command Classes.

CC:0070.02.00.11.003
A device MUST be able to operate with default factory configuration parameter values.

CC:0070.02.00.12.001
It is RECOMMENDED that configuration parameters can be manipulated via a local user interface.

CC:0070.02.00.12.002
It is RECOMMENDED that default factory configuration parameter values can be restored via a local user interface.

4.31.1 Compatibility considerations

Configuration Command Class, version 2 extends the addressing space to 65.535 product specific configuration parameters. Further, Configuration Command Class, version 2 makes it possible to set multiple configuration parameters with one Command.

CC:0070.02.00.21.001
A device supporting Configuration Command Class, version 2 MUST support Configuration Command Class, version 1.

4.31.1.1 "Default" flag

Refer to 4.30.1.1

4.31.2 Interoperability considerations

Configuration Command Class, version 2 is backwards compatible with Configuration Command Class, version 1.

CC:0070.02.00.31.001
Configuration Command Class, version 2 does not extend any existing commands of Configuration Command Class, version 1. The 255 parameters that can be addressed by the Configuration Set Command MUST be identical to the first 255 parameters that can be addressed by the (version 2) Configuration Bulk Set Command.

4.31.3 Configuration Set Command

This Command is used to set the value of a configuration parameter.

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 N							

Parameter Number (8 bits)

CC:0070.02.04.13.001
CC:0070.02.04.12.001
This field is used to specify the actual configuration parameter. Parameter Numbers MAY be product specific. Parameter Numbers SHOULD be assigned in a sequence starting from 1.

Default (1 bit)

This field is used to specify if the default value is to be restored for the specified configuration parameter.

CC:0070.02.04.11.001
CC:0070.02.04.11.002
The value 1 MUST indicate that default factory settings must be restored for the specified Parameter Number. In this case, the Configuration Value field MUST be ignored and the receiving node MUST reset the specified parameter and SHOULD NOT reset any other parameters.

CC:0070.02.04.11.003
The value 0 MUST indicate that the specified Parameter Number must assume the value specified by the Configuration Value field.

Reserved

CC:0070.02.04.11.004
This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Size (3 bits)

CC:0070.02.04.11.005
This field is used to specify the size of the actual parameter. The advertised size of a given parameter MUST always be the same. The value of this field MUST comply with Table 32.

CC:0070.02.04.11.006
All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

Configuration Value (N bytes)

CC:0070.02.04.11.007
This field carries the value to be assigned. The size of the field MUST comply with the size advertised by the Size field.

CC:0070.02.04.11.008
The field MUST carry a signed value. The binary encoding of the signed value MUST comply withTable 10. The field Value 1 MUST be the most significant byte.

4.31.4 Configuration Bulk Set Command

This command is used to set the value of one or more configuration parameters.

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

Parameter Offset (16 bits)

CC:0070.02.07.11.001 This field is used to specify the first parameter in a range of one or more parameters. The first byte MUST carry the most significant byte of the 16 bit value.

Number of Parameters (8 bits)

This field is used to specify the number (M) of configuration parameters contained in this command.

CC:0070.02.07.11.002 The value of this field MUST be in the range 1..255.

CC:0070.02.07.11.010 The advertised parameters MUST be in a consecutive range. For example, Parameter Offset set to 4 and Number of Parameters set to 3 MUST represent parameters number 4, 5 and 6.

CC:0070.02.07.11.011 A receiving node MUST ignore values for non-existing parameters

Default (1 bit)

This field is used to specify if the default value is to be restored for the specified configuration parameters.

CC:0070.02.07.11.003 The value 1 MUST indicate that default factory settings must be restored for the Parameter Numbers defined by the Parameter Offset and Number of Parameters fields. In this case, the Configuration Value field MUST be ignored and the receiving node MUST reset the specified parameters and SHOULD NOT reset any other parameters.

CC:0070.02.07.11.004

CC:0070.02.07.11.005 The value 0 MUST indicate that the specified Parameter Numbers must assume the value specified by the Configuration Value field.

Handshake (1 bit)

This field is used to indicate if a Configuration Bulk Report Command is to be returned when the specified configuration parameters have been stored in non-volatile memory.

CC:0070.02.07.11.006
CC:0070.02.07.11.007
CC:0070.02.07.11.008
CC:0070.02.07.11.009
CC:0070.02.07.12.001 If the Handshake bit is set to 1, a Configuration Bulk Report Command MUST be returned. The command MUST echo all parameters found in this Configuration Bulk Set Command. A node returning a Configuration Bulk Report Command with the Handshake bit set MUST be ready to receive another Configuration Bulk Set Command. If the Handshake bit is set, an originating node MUST wait for the Configuration Bulk Report Command for at least one second. If not receiving a report, the originating node SHOULD assume that the operation failed and resend the same Configuration Bulk Set Command one more time.

CC:0070.02.07.11.00A If the Handshake bit is set to 0, a receiving node MUST NOT return a Configuration Bulk Report in response.

Reserved

CC:0070.02.07.11.00B This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Size (3 bits)

CC:0070.02.07.11.00C This field is used to specify the size of the actual parameters. The advertised size of a given parameter MUST always be the same. The value of this field MUST comply with Table 32.

CC:0070.02.07.11.00D All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

Parameter – Configuration Value (M*N bytes)

CC:0070.02.07.11.00E These fields carry the values to be assigned. Each field MUST have the same size. The size of each field MUST comply with the size advertised by the Size field.

CC:0070.02.07.11.00F The field MUST carry a signed value. The binary encoding of the signed value MUST comply withTable 10. The Value 1 field MUST be the most significant byte.

4.31.5 Configuration Bulk Get Command

This command is used to query the value of one or more configuration parameters.

CC:0070.02.08.11.001 The Configuration Bulk Report Command MUST be returned in response to this command.

CC:0070.02.08.11.002 This command MUST NOT be issued via multicast addressing.

CC:0070.02.08.11.003 A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

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

Parameter Offset (16 bits)

CC:0070.02.08.11.004 This field is used to specify the first parameter in a range of one or more parameters. The first byte MUST carry the most significant byte of the 16 bit value.

CC:0070.02.08.12.001 Parameter Numbers SHOULD be assigned in a sequence starting from 1.

Number of Parameters (8 bits)

This field is used to specify the number of requested configuration parameters.

CC:0070.02.08.11.005 The value of this field MUST be in the range 1..255.

CC:0070.02.08.11.006 The advertised parameters MUST be in a consecutive range. For example, Parameter Offset set to 4 and Number of Parameters set to 3 MUST represent parameters number 4, 5 and 6.

4.31.6 Configuration Bulk Report Command

This command is used to advertise the actual value of one or more advertised parameters.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_CONFIGURATION							
Command = CONFIGURATION_BULK_REPORT							
Parameter Offset 1 (MSB)							
Parameter Offset 2 (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 M – Configuration Value 1 (MSB)							
…							
Parameter M – Configuration Value N (LSB)							

Parameter Offset (16 bits)

CC:0070.02.09.11.001 This field is used to advertise the first parameter in a range of one or more parameters. The first byte MUST carry the most significant byte of the 16 bit value.

Number of Parameters (8 bits)

This field is used to advertise the number (M) of configuration parameters contained in this command.

CC:0070.02.09.11.002 The value of this field MUST be in the range 1..255.

CC:0070.02.09.11.012 The advertised parameters MUST be in a consecutive range. For example, Parameter Offset set to 4 and Number of Parameters set to 3 MUST represent parameters number 4, 5 and 6, regardless of whether they exist at the sending node.

Reports to follow (8 bits)

CC:0070.02.09.11.003 This field MUST be used to advertise the number of reports left before all requested configuration parameters values have been transferred.

CC:0070.02.09.11.004 The value 0 MUST indicate that this is the last report.

CC:0070.02.09.11.005 If the Handshake field is 1 , this field MUST be set to 0.

CC:0070.02.09.11.006 The value of this field MUST be in the range 0..255.

Default (1 bit)

CC:0070.02.09.11.007 This field MUST be used to advertise if all advertised configuration parameters have the factory default value.

CC:0070.02.09.11.008 The value 1 MUST indicate that all advertised configuration parameters have the factory default value.
CC:0070.02.09.11.009 The value 0 MUST indicate that one or more of the advertised configuration parameters do not have the factory default value.

Handshake (1 bit)

This field is used to indicate if this report is returned in response to a Configuration Bulk Set Command.

CC:0070.02.09.11.00A The value 1 MUST indicate that all configuration parameters have been stored in non-volatile memory and that the sending node is ready to receive another Configuration Bulk Set Command. Except for the
CC:0070.02.09.11.00B Reports to Follow field, all other fields MUST echo the values received in the Configuration Bulk Set
CC:0070.02.09.11.00C Command. The Reports to Follow field MUST be 0.

CC:0070.02.09.11.00D The value 0 MUST indicate that this report is returned in response to a Configuration Bulk Get Command.

Size (3 bits)

CC:0070.02.09.11.00E This field is used to advertise the size of the actual parameter. The advertised size of a given parameter MUST always be the same. The value of this field MUST comply with Table 32.

CC:0070.02.09.11.00F All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

Parameter – Configuration Value (M*N bytes)

CC:0070.02.09.11.010 These fields carry the values advertised by the Parameter Offset and Number of Parameters fields. Each field MUST have the same size. The size of each field MUST comply with the size advertised by the Size field.

CC:0070.02.09.11.011 The field MUST carry a signed value. The binary encoding of the signed value MUST comply withTable 10. The Value 1 field MUST be the most significant byte.

CC:0070.02.09.12.001 A sending node SHOULD set this field to 0 for non-existing parameters.

4.32 Configuration Command Class, version 3

The Configuration Command Class, version 3 allows product specific configuration parameters to be changed. Examples include the default dimming rate of a light dimmer and the endpoints of a window covering device.

CC:0070.03.00.11.001 Configuration parameters accessed via this command class MUST NOT replace similar commands provided by other existing application Command Classes.

CC:0070.03.00.11.002 Configuration parameter documentation MUST be provided in the product documentation.

CC:0070.03.00.12.001 Configuration parameter documentation SHOULD be provided via Z-Wave via this command class version.

CC:0070.03.00.11.003 A device MUST be able to operate with default factory configuration parameter values.

CC:0070.03.00.12.002 It is RECOMMENDED that configuration parameters can be manipulated via a local user interface.

CC:0070.03.00.12.003 It is RECOMMENDED that default factory configuration parameter values can be restored via a local user interface.

4.32.1 Compatibility considerations

The following additions are made to the Configuration Command Class, version 3:

- Configuration Name Get Command
- Configuration Name Report Command
- Configuration Info Get Command
- Configuration Info Report Command
- Configuration Properties Get Command
- Configuration Properties Report Command

The purpose of the above commands is to allow a manufacturer to advertise information relating to the use of a specific configuration parameter directly from the configuration interface. The advertised information should be the same as can be found in the printed documentation.

It should be emphasized that, just like versions 1 and 2, the Configuration Command Class, version 3 is intended only for access to product specific configuration parameters. The access to parameter information is only meant to facilitate access via Z-Wave to information which can also be found in the printed documentation.

Configuration Command Class versions 1 and 2 explicitly require that the binary value of all configuration parameters are represented as signed values encoded as binary 2's complement.

Configuration Command Class v3 allows for bit field, signed integer or unsigned integer representations of each individual parameter.

CC:0070.03.00.21.001 A device supporting the Configuration Command Class v3 MUST respond to a Configuration Properties Get command. If a Configuration Properties Report command is not returned in response to a

CC:0070.03.00.21.002 Configuration Properties Get command, a controlling device MUST treat the parameter value as a signed integer.

Configuration Command Class version 2 extended the possible parameter range from 1..255 to 1..65535. The first 255 parameters of the two ranges are identical. Parameters in the range 1..255 can

be addressed using either the Bulk Set/Get Commands or Set/Get Commands while parameters in the range 256..65535 can only be addressed using the Bulk Set/Get Commands.

4.32.1.1 "Default" flag

Refer to 4.30.1.1.

4.32.1.2 Configuration Properties Report

CC:0070.03.00.21.003 Earlier text revisions of the Configuration Command Class, version 3 did not explicitly specify that the "Min Value", "Max Value" and "Default Value" fields MUST be omitted if the advertised size is 0 in the Configuration Properties Report Command.

CC:0070.03.00.22.001 Controlling nodes SHOULD be aware that some legacy nodes supporting version 3 could by error include the "Min Value", "Max Value" and "Default Value" fields and set them to 0x00 with an arbitrary

CC:0070.03.00.22.002 size. If a controlling node receives a Report in which the "Next Parameter Number" field seems to be set to 0x0000 when requesting parameter number 0, the controlling node SHOULD inspect the last 2 bytes of the command frame in order to find out what is the Next Parameter Number.

4.32.2 Interoperability considerations

CC:0070.03.00.32.001 It is RECOMMENDED that the name and information stored in a supporting device are in English.

CC:0070.03.00.32.002 It is RECOMMENDED that the last line of the information field presents an Internet URL which points to updated information for the actual configuration parameter as well as other language variants.

4.32.3 Configuration Name Get Command

This command is used to request the name of a configuration parameter.

CC:0070.03.0A.11.001 The Configuration Name Report Command MUST be returned in response to this command.

CC:0070.03.0A.11.002 This command MUST NOT be issued via multicast addressing.

CC:0070.03.0A.11.003 A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_CONFIGURATION							
Command = CONFIGURATION_NAME_GET							
Parameter Number 1 (MSB)							
Parameter Number 2 (LSB)							

Parameter Number (16 bits)

This field is used to specify the requested configuration parameter.

CC:0070.03.0A.11.004 The first byte MUST carry the most significant byte of the 16 bit value.

4.32.4 Configuration Name Report Command

This command is used to advertise the name of a parameter.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_CONFIGURATION							
Command = CONFIGURATION_NAME_REPORT							
Parameter Number 1 (MSB)							
Parameter Number 2 (LSB)							
Reports to follow							
Name 1							
..							
Name N							

Parameter Number (16 bits)

This field is used to advertise the actual configuration parameter.

CC:0070.03.0B.11.001 The first byte MUST carry the most significant byte of the 16 bit value.

CC:0070.03.0B.13.001 If a non-existing parameter is specified in a Configuration Name Get Command, a receiving node MAY return a zero-length Name field in the Configuration Name Report Command. The receiving node SHOULD however return an error message e.g. “Unassigned parameter”.

CC:0070.03.0B.12.001

Parameter numbers above 255 can only be addressed using the Bulk Set/Get Commands. Parameter numbers in the range 1..255 can be addressed using either the Bulk Set/Get Commands or the Set/Get Commands.

Reports to follow (8 bits)

This field is used to advertise the number of reports left before all parts of the command have been transferred.

CC:0070.03.0B.11.002 The value 0 MUST indicate that this is the last report.

Name (N bytes)

CC:0070.03.0B.12.002 This field is used to advertise the name of the parameter. It is RECOMMENDED that the Name field advertises the name found in the documentation.

CC:0070.03.0B.11.003 The field MUST carry a byte string with no zero termination. The Name field MAY span multiple reports. The number of Name bytes transmitted MUST be determined from the length field in the frame.

CC:0070.03.0B.11.004 Bytes MUST be encoded in UTF-8 format.

CC:0070.03.0B.12.003 It is RECOMMENDED that the name stored in a supporting device is in English.

4.32.5 Configuration Info Get Command

This command is used to request usage information for a configuration parameter.

CC:0070.03.0C.11.001 The Configuration Info Report Command MUST be returned in response to this command.

CC:0070.03.0C.11.002 This command MUST NOT be issued via multicast addressing.

CC:0070.03.0C.11.003 A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_CONFIGURATION							
Command = CONFIGURATION_INFO_GET							
Parameter Number 1 (MSB)							
Parameter Number 2 (LSB)							

Parameter Number (16 bits)

This field is used to specify the requested configuration parameter.

CC:0070.03.0C.11.004 The first byte MUST carry the most significant byte of the 16 bit value.

4.32.6 Configuration Info Report Command

This command is used to advertise usage information for a configuration parameter.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_CONFIGURATION							
Command = CONFIGURATION_INFO_REPORT							
Parameter Number 1 (MSB)							
Parameter Number 2 (LSB)							
Reports to follow							
Info 1							
…							
Info N							

Parameter Number (16 bits)

This field is used to advertise the actual configuration parameter.

CC:0070.03.0D.11.001 The first byte MUST carry the most significant byte of the 16 bit value.

CC:0070.03.0D.13.001
CC:0070.03.0D.12.001 If a non-existing parameter is specified in a Configuration Info Get Command, a receiving node MAY return a zero-length Info field in the Configuration Info Report Command. The receiving node SHOULD however return an error message, e.g. “Unassigned parameter”.

Parameter numbers above 255 can only be addressed using the Bulk Set/Get Commands. Parameter numbers in the range 1..255 can be addressed using either the Bulk Set/Get Commands or the Set/Get Commands

Reports to follow (8 bits)

This field is used to advertise the number of reports left before all parts of the command have been transferred.

CC:0070.03.0D.11.002 The value 0 MUST indicate that this is the last report.

Info (N bytes)

CC:0070.03.0D.12.002 This field is used to advertise the detailed information available for the parameter. It is RECOMMENDED that the Info field carries all information found in the documentation.

CC:0070.03.0D.11.003 The field MUST carry a byte string with no zero termination.

CC:0070.03.0D.11.004 The number of Info bytes transmitted MUST be determined from the length field in the frame.

CC:0070.03.0D.13.002 The Info field MAY span multiple reports. The number of Info bytes MAY be zero.

CC:0070.03.0D.11.005 Bytes MUST be encoded in UTF-8 format.

CC:0070.03.0D.12.003 It is RECOMMENDED that the information stored in a supporting device is in English.

CC:0070.03.0D.12.004 It is RECOMMENDED that the last line of the information field presents an Internet URL which points to updated information for the actual configuration parameter as well as other language variants.

4.32.7 Configuration Properties Get Command

This command is used to request the properties of a configuration parameter.

The Configuration Properties Report Command MUST be returned in response to this command.

If a Configuration Properties Report command is not returned in response to a Configuration Properties Get command, a controlling device MUST treat the parameter value as a signed integer.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_CONFIGURATION							
Command = CONFIGURATION_PROPERTIES_GET							
Parameter Number 1 (MSB)							
Parameter Number 2 (LSB)							

Parameter Number (16 bits)

This field is used to specify the requested configuration parameter.

CC:0070.03.0E.11.001 The first byte MUST carry the most significant byte of the 16 bit value.

CC:0070.03.0E.11.002
CC:0070.03.0E.11.003 If a non-existing parameter is specified in this command, a receiving node MUST advertise zero values in the Format and Size fields. The Next Parameter Number field MUST advertise the next available configuration parameter.

CC:0070.03.0E.12.001 It is RECOMMENDED that a controlling device initiates probing of supported configuration parameters by issuing this command for parameter number 0.

CC:0070.03.0E.12.002 If a Size field value of zero is returned, a controlling device SHOULD issue a Configuration Properties Get Command for the parameter advertised in the Next Parameter Number field.

4.32.8 Configuration Properties Report Command

This command is used to advertise the properties of a configuration parameter.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_CONFIGURATION							
Command = CONFIGURATION_PROPERTIES_REPORT							
Parameter Number 1 (MSB)							
Parameter Number 2 (LSB)							
Reserved			Format			Size	
Min Value 1 (MSB)							
..							
Min Value N (LSB)							
Max Value 1 (MSB)							
..							
Max Value N (LSB)							
Default Value 1 (MSB)							
..							
Default Value N (LSB)							
Next Parameter Number 1 (MSB)							
Next Parameter Number 2 (LSB)							

Parameter Number (16 bits)

This field is used to advertise the actual configuration parameter.

CC:0070.03.0F.11.001 The first byte MUST carry the most significant byte of the 16 bit value.

CC:0070.03.0F.11.002
CC:0070.03.0F.11.003 If this field advertises a non-existing parameter , a receiving node MUST advertise zero values in the Format and Size fields. The Next Parameter Number field MUST advertise the next available configuration parameter.

Parameter numbers above 255 can only be addressed using the Bulk Set/Get Commands. Parameter numbers in the range 1..255 can be addressed using either the Bulk Set/Get Commands or the Set/Get Commands.

Reserved

CC:0070.03.0F.11.004 This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Format (3 bits)

This field is used to advertise the format of the parameter.

CC:0070.03.0F.11.005 The value of this field MUST comply with Table 33.

Table 33, Configuration Properties Report :: Format encoding

Value	Parameter format and presentation
0x00	Signed Integer
0x01	Unsigned Integer
0x02	Enumerated (Radio buttons)
0x03	Bit field (Checkboxes)

CC:0070.03.0F.11.006 All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

CC:0070.03.0F.11.007 If the parameter format is “Unsigned integer”, normal binary integer encoding MUST be used.

CC:0070.03.0F.11.008 If the parameter format is “Signed integer”, the binary encoding MUST comply with Table 10.

CC:0070.03.0F.11.009 If the parameter format is “Enumerated”, the parameter MUST be treated as an unsigned integer.
CC:0070.03.0F.12.001 A graphical configuration tool SHOULD present this parameter as a series of radio buttons [11].

CC:0070.03.0F.11.00A If the parameter format is “Bit field” the parameter MUST be treated as a bit field where each individual bit can be set or reset.
CC:0070.03.0F.12.002 A graphical configuration tool SHOULD present this parameter as a series of checkboxes [11].

Size (3 bits)

This field is used to advertise the size of the actual parameter.

CC:0070.03.0F.11.00B The advertised size MUST also apply to the fields “Min Value”, “Max Value”, “Default Value” carried in this command.

CC:0070.03.0F.11.00C The value of this field MUST comply with Table 34.

Table 34, Configuration Properties Report :: Size encoding

Size	Size of parameter
0	0 bit (Unassigned parameter, Value fields omitted)
1	8 bit
2	16 bit
4	32 bit

CC:0070.03.0F.11.00D All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

Min Value (N bytes)

CC:0070.03.0F.11.00E This field MUST advertise the minimum value that the actual parameter can assume.

CC:0070.03.0F.11.00F If the parameter is “Bit field”, this field MUST be set to 0.

CC:0070.03.0F.11.010 The size of the field MUST comply with the size advertised by the Size field. This field MUST be omitted if the Size field is set to 0.

CC:0070.03.0F.11.011 The sign encoding MUST comply with the mode advertised by the Signed field.

Max Value (N bytes)

CC:0070.03.0F.11.012 This field MUST advertise the maximum value that the actual parameter can assume.

CC:0070.03.0F.11.013 If the parameter is “Bit field”, each individual supported bit MUST be set to ‘1’, while each un-supported bit of MUST be set to ‘0’.

CC:0070.03.0F.12.003 A graphical configuration tool SHOULD NOT present checkboxes for un-supported bits.

CC:0070.03.0F.11.014 The size of the field MUST comply with the size advertised by the Size field. This field MUST be omitted if the Size field is set to 0.

CC:0070.03.0F.11.015 The sign encoding MUST comply with the mode advertised by the Signed field.

Default Value (N bytes)

CC:0070.03.0F.11.016 This field MUST advertise the default value of the actual parameter.

CC:0070.03.0F.11.017 The size of the field MUST comply with the size advertised by the Size field. This field MUST be omitted if the Size field is set to 0.

CC:0070.03.0F.11.018 The sign encoding MUST comply with the mode advertised by the Signed field.

Next Parameter Number (16 bits)

Configuration parameter identifiers may be assigned in a non-sequential order.

CC:0070.03.0F.11.019 This field MUST advertise the next available configuration parameter.
The value 0x0000 MUST indicate that this is the last available configuration parameter.

CC:0070.03.0F.11.01A The first byte MUST carry the most significant byte of the 16 bit value.

4.33 Configuration Command Class, version 4

The Configuration Command Class, version 4 introduces the support of read-only parameters, parameters requiring network re-inclusion and advanced parameters as well as allows optional support for Bulk Commands.

A read-only parameter can be accessed with Get commands but cannot be modified with Set Commands. A parameter requiring re-inclusion is typically a parameter altering the node capabilities, which requires performing the inclusion again in order to have the controller to learn about the node capabilities once more. A parameter may be advertised as "advanced" to indicate to the controlling application that the actual parameter is intended only for advanced use, e.g. calibration.

The following commands are modified in version 4:

- Configuration Set Command
- Configuration Bulk Set Command
- Configuration Properties Report Command

The following new command is introduced:

- Configuration Default Reset Command

All commands not mentioned in this version remain unchanged from previous versions.

4.33.1 Compatibility considerations

Configuration Command Class, version 4 is backwards compatible with the previous versions of Configuration Command Class.

CC:0070.04.00.21.001 A device supporting Configuration Command Class, version 4 MUST support Configuration Command Class, version 3.

4.33.1.1 Multi Channel Consideration

CC:0070.04.00.12.001 Multi Channel End Points SHOULD NOT support the Configuration Command Class.

4.33.1.2 “Default” flag

Refer to 4.30.1.1

CC:0070.04.00.21.002 From version 4 and onwards, a node MUST NOT reset all Configuration Parameters when receiving a Set or Bulk Set Command with the default flag set to 1.

4.33.1.3 Configuration Properties Report

Refer to 4.32.1.2.

The recommendation given in version 3 does not apply to version 4 supporting nodes as new fields are appended at the end of the Configuration Properties Report Command in version 4.

4.33.1.4 "Altering capabilities" flag

CC:0070.04.00.21.003 Configuration parameters modifying a node's and/or (non-dynamic) Multi Channel End Point capabilities MUST be advertised as "Altering capabilities".

CC:0070.04.00.21.004 When such a parameter is modified:

- A Z-Wave or Z-Wave Plus node MUST NOT advertise the new and/or Multi Channel End Point capabilities before being excluded from its current network.
- A Z-Wave Plus version 2 node MUST advertise its new capabilities immediately.

4.33.1.5 "Advanced" flag

The difference between normal and advanced parameters lies on the controlling node side.

CC:0070.04.00.23.001 A parameter MAY be advertised as "advanced" in order to simplify the configuration of a node by normally not showing such a parameter to the end user.

CC:0070.04.00.22.001 Advanced parameters SHOULD be presented to the user only when an [Advanced] option is selected in the controller user interface.

4.33.1.6 Parameters value and network inclusion/exclusion

CC:0070.04.00.21.005 From version 4 and onwards, a node MUST NOT modify or reset any configuration parameter when being included or excluded of a Z-Wave network.

CC:0070.04.00.21.006 A node MUST reset all its configuration parameters if either:

- It is manually reset to factory default
- It receives a Configuration Default Reset Command.

CC:0070.04.00.21.007 A node MUST NOT reset all its configuration parameters in any other case.

4.33.1.7 Bulk commands support

CC:0070.04.00.22.002 A node supporting Configuration Command Class, version 4 MAY elect to ignore the following Commands:

- Configuration Bulk Set Command
- Configuration Bulk Get Command
- Configuration Bulk Report Command.

CC:0070.04.00.21.008 If it is the case, the node MUST:

- Return an Application Rejected Request Command when receiving one of the ignored commands (if received without Supervision encapsulation).
- Advertise that it ignores the Bulk Commands in the Configuration Properties Report.

CC:0070.04.00.21.009 If Bulk Commands are supported, they MUST be supported for all parameters. If Bulk Commands are ignored, they MUST be ignored for all parameters.

4.33.2 Configuration Set Command

This command is used to set the value of a configuration parameter.

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 N							

CC:0070.04.04.11.001 A receiving node MUST ignore this command if the specified parameter is advertised as a Read-only parameter.

All fields not specified below are unchanged from version 2 Configuration Set Command.

Default (1 bit)

This field is used to specify if the default value is to be restored for the specified configuration parameter.

CC:0070.04.04.11.002
CC:0070.04.04.11.003 The value 1 MUST indicate that default factory settings must be restored for the specified Parameter Number. In this case, the Configuration value field MUST be ignored and any other parameters MUST NOT be reset by a receiving node.

CC:0070.04.04.11.004 The value 0 MUST indicate that the specified Parameter Number MUST assume the value specified by the Configuration Value field.

Configuration Value (N bytes)

CC:0070.04.04.11.005 This field carries the value to be assigned. The size of the field MUST comply with the size advertised by the Size field.

CC:0070.04.04.11.006 The configuration value MUST be encoded according to the Format field advertised in the Configuration Properties Report Command for the parameter number.

4.33.3 Configuration Bulk Set Command

This command is used to set the value of one or more configuration parameters.

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

CC:0070.04.07.11.001 A receiving node MUST ignore this command for parameters that are advertised as Read-only parameters.

All fields not specified below are unchanged from version 2 Configuration Bulk Set Command.

Default (1 bit)

This field is used to specify if the default value is to be restored for the specified configuration parameters.

CC:0070.04.07.11.002 The value 1 MUST indicate that default factory settings must be restored for all the Parameter Numbers defined by the Parameter Offset and Number of Parameters fields. In this case, the Configuration Value field MUST be ignored and any other parameters MUST NOT be reset by a receiving node.

CC:0070.04.07.11.003

CC:0070.04.07.11.004 The value 0 MUST indicate that the specified Parameter Numbers MUST assume the value specified by the Configuration Value field.

Parameter –Configuration Value (M*N bytes)

CC:0070.04.07.11.005 These fields carry the parameter values to be assigned. Each parameter value field MUST have the same size. The size of each field MUST comply with the size advertised by the Size field.

CC:0070.04.07.11.006 The values MUST be encoded according to the Format field advertised in the Configuration Properties Report Command for each parameter number.

4.33.4 Configuration Properties Report Command

This command is used to advertise the properties of a configuration parameter.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_CONFIGURATION							
Command = CONFIGURATION_PROPERTIES_REPORT							
Parameter Number 1 (MSB)							
Parameter Number 2 (LSB)							
Altering capabilities	Read-only	Format			Size		
Min Value 1 (MSB)							
..							
Min Value N (LSB)							
Max Value 1 (MSB)							
..							
Max Value N (LSB)							
Default Value 1 (MSB)							
..							
Default Value N (LSB)							
Next Parameter Number 1 (MSB)							
Next Parameter Number 2 (LSB)							
Reserved						No Bulk support	Advanced

All fields not specified below are unchanged from version 3 Configuration Properties Report Command.

Read-only (1 bit)

This field is used to indicate if the parameter is read-only.

CC:0070.04.0F.11.001 The value 1 MUST indicate that the advertised parameter is read-only.

CC:0070.04.0F.11.002 The value 0 MUST indicate that the advertised parameter is not read-only.

Altering capabilities (1 bit)

This field is used to indicate if the advertised parameter triggers a change in the node's capabilities.

CC:0070.04.0F.11.003 The value 1 MUST indicate that the advertised parameter alters the node capabilities when being changed.

CC:0070.04.0F.11.004 The value 0 MUST indicate that the advertised parameter does not alter the node capabilities when being changed.

Refer to 4.33.1.4 "Altering capabilities" flag.

Advanced (1 bit)

This field is used to indicate if the advertised parameter is to be presented in the "Advanced" parameter section in the controller GUI.

CC:0070.04.0F.11.005 The value 1 MUST indicate that the advertised parameter is an Advanced parameter.

CC:0070.04.0F.11.006 The value 0 MUST indicate that the advertised parameter is not an Advanced parameter.

No Bulk support (1 bit)

This field is used to advertise if the sending node supports Bulk Commands.

CC:0070.04.0F.11.007 The value 1 MUST indicate that the Bulk Commands will be ignored by the sending node.

CC:0070.04.0F.11.008 The value 0 MUST indicate that the Bulk Commands are supported by the sending node

CC:0070.04.0F.11.009 A sending node MUST always advertise the same value in this field, regardless of the parameter number.

4.33.5 Configuration Default Reset Command

This command is used to reset all configuration parameters to their default value.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_CONFIGURATION							
Command = CONFIGURATION_DEFAULT_RESET							

CC:0070.04.01.11.001 A node receiving this command MUST reset all its Configuration Parameters to their default value.

4.34 Controller Replication Command Class, version 1

The Controller Replication Command Class is used to copy scene and group data to another controlling node. The Command Class may be used in conjunction with a controller shift or when including a new controller to the network. It is OPTIONAL to use this command class during a controller shift or when including a new controller to the network.

4.34.1 Transfer Group Command

This command is 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)

Sequence Number of this particular command.

Group ID (8 bits)

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

Node ID (8 bits)

Node ID of slave device.

4.34.2 Transfer Group Name Command

This command is 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 N							

Sequence Number (8 bits)

Sequence Number of this particular command.

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.

4.34.3 Transfer Scene Command

This command is 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)

Sequence Number of this particular command.

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.

4.34.4 Transfer Scene Name Command

This command is 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 N							

Sequence Number (8 bits)

Sequence Number of this particular command.

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.

4.35 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 optionally supported at different security levels. (E.g. DCP monitoring commands could be supported using non-secure communication, while enabling strict secure-only communication for the DCP Configuration Command Class). Refer to the Security and Security 2 command classes for more details.

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, refer to the Rate Table Configuration Command Class) is associated with the event. When a DCP event is outdated, it is 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.

4.35.1 DCP List Supported Get Command

This command is used to request the total size of the DCP list along with the number of free entries in the list.

The DCP List Supported Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

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

4.35.2 DCP List Supported Report Command

This command is 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 bits)

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

Free DCP List entries (8 bits)

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

4.35.3 DCP List Set Command

This command is 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

Timestamp -Year (16 bits)

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

Timestamp -Month (8 bits)

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

Timestamp -Day (8 bits)

Specify the day of the month between 01 and 31.

Timestamp -Hour Local Time (8 bits)

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

Timestamp -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.

Timestamp -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.

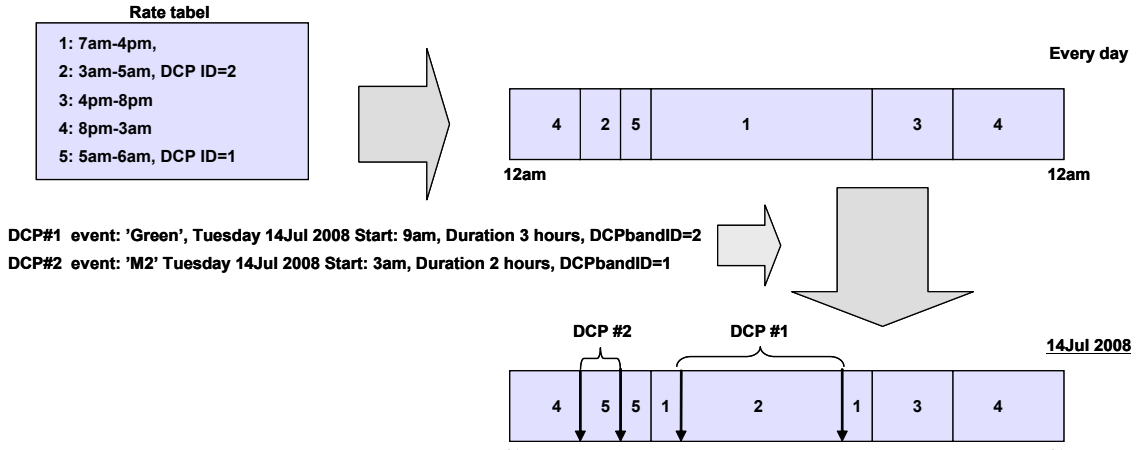
DCP Rate ID (8 bits)

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.

DCP Rate ID usage Example

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.



Randomization interval (8 bits)

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 bits)

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

Generic Device Class (8bits)

Specify the Generic Device Class identifier. Refer to [10] and [22].

Specific Device Class (8 bits)

Specify the Specific Device Class identifier. Refer to [10] and [22].

Start Year (16 bits)

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 bits)

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

Start Day (8 bits)

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

Start Hour Local Time (8 bits)

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 bits)

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 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.

Duration Hour Time (8 bits)

Specify the number of complete hours of the event.

Duration Minute Time (8 bits)

Specify the number of complete minutes of the event.

Duration Second Time (8 bits)

Specify the number of complete seconds of the event.

Event Priority (8 bits)

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

All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

Load shedding (8 bits)

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.

This field MUST be in the range 0x01..0x64, which represents values in the range 1%..100%. All other values are reserved. Reserved values MUST NOT be used by a sending node and MUST be ignored by a receiving node.

Start Association group (8 bits)

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 bits)

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^{o}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.

4.35.4 DCP List Remove

This command is 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							

Refer to DCP List report command class (section 4.36.2) for description of fields

4.36 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 Rate ID, refer to the Rate Table Configuration Command Class) is associated with the event. When a DCP event is outdated, it is removed from the list. Each DCP event is uniquely identified by a timestamp issued the Utility Supplier.

4.36.1 DCP List Get Command

This command is used to request the pending DCP event in a device.

The DCP List Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

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

4.36.2 DCP List Report Command

This 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.

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 bits)

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.

Refer to DCP List Set command class (section 4.35.3) for detailed description of the fields.

4.36.3 DCP Event Status Get

This command is used to query the status of a specific DCP event in the DCP list.

The DCP Event Status Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

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							

Refer to DCP List Set command class (section 4.35.3) for detailed description of the fields.

4.36.4 DCP Event Status Report

This command is 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							

Refer to DCP List report command class (section 4.36.2) for description of fields

Event Status (8 bits)

The field contains the status of the event.

Table 35, DCP Event Status Report::Event Status encoding

Event status	Description
0x01	Event Started
0x02	Event Completed
0x03	Event Rejected by the user
0x04	Event not Applicable

All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

4.37 Door Lock Command Class, version 1-2

The Door Lock Command Class is used to operate and configure a door lock device.

The Door Lock Command Class is an actuator control command class. Refer to 3.6.

4.37.1 Compatibility considerations

A device supporting Door Lock CC, Version 2 MUST support Door Lock CC, version 1.

CC:0062.02.00.21.001 The Door Lock Command Class, version 2 adds the “unknown” state to the Door Lock Operation Report Command.

CC:0062.01.00.21.001 A supporting node MAY implement a subset of the features represented by the Door Lock Mode, Door Handles Mode and Door Condition fields which are provided by the commands of this Command Class.

4.37.2 Door Lock Operation Set Command

This command is used to set the operation mode of a supporting 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)

The Door Lock Mode field is used to specify the operation mode of the door lock device.

CC:0062.01.01.11.001 The encoding of this field MUST be according to Table 36.

Table 36, Door Lock Operation Set :: Mode

Mode	Description	Version
0x00	Door Unsecured 1)	1
0x01	Door Unsecured with timeout 2)	1
0x10	Door Unsecured for inside Door Handles 1)	1
0x11	Door Unsecured for inside Door Handles with timeout 2)	1
0x20	Door Unsecured for outside Door Handles 1)	1
0x21	Door Unsecured for outside Door Handles with timeout 2)	1
0xFF	Door Secured	1

1) Constant mode. Door will be unsecured until set to secured mode by another command.
2) Timeout mode. Fallback to secured mode after timeout has expired.

CC:0062.01.01.11.002 All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

CC:0062.01.01.11.003 A controlling node MUST NOT specify modes using timeout-based fall back if the Operation Type of the supporting device is set to Constant operation (set by Door Lock Configuration Set).

CC:0062.01.01.11.004 A supporting node MUST ignore modes using timeout-based fall back if Operation Type is set to Constant operation.

CC:0062.01.01.11.006 A supporting node MUST apply the constant modes without any timeout even if it is set to Timed Operation Type.

CC:0062.01.01.13.001 A supporting node MAY implement a subset of the Door Lock Modes defined by Table 36.

CC:0062.01.01.11.005 A supporting node MUST accept the Door Lock Mode values 0x00 and 0xFF.

4.37.3 Door Lock Operation Get Command

This command is used to request the status of a door lock device.

CC:0062.01.02.11.001 The Door Lock Operation Report command MUST be returned in response to this command.

CC:0062.01.02.11.002 This command MUST NOT be issued via multicast addressing.

CC:0062.01.02.11.003 A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

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

4.37.4 Door Lock Operation Report Command

This command is used to advertise the status of a door lock device.

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)

CC:0062.01.03.11.001 The Door Lock Mode field MUST advertise the mode of the door lock device.

CC:0062.01.03.11.002 The encoding of this field MUST be according to Table 37.

Table 37, Door Lock Operation Report :: Door Lock Mode

Mode	Description	Version
0x00	Door Unsecured 1)	1
0x01	Door Unsecured with timeout 2)	1
0x10	Door Unsecured for inside Door Handles 1)	1
0x11	Door Unsecured for inside Door Handles with timeout 2)	1
0x20	Door Unsecured for outside Door Handles 1)	1
0x21	Door Unsecured for outside Door Handles with timeout 2)	1
0xFE	Door/Lock State Unknown 3)	2
0xFF	Door Secured	1

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

3) Bolt is not fully retracted/engaged

CC:0062.01.03.11.003 All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

CC:0062.01.03.13.001 A supporing node MAY advertise a subset of the available Door Lock Mode values.

CC:0062.01.03.11.004 A controlling node MUST accept all Door Lock Mode values.

CC:0062.01.03.12.001 The Door Lock Mode field SHOULD advertise the current value of the device hardware; also while in transition to a new target value.

Outside Door Handles Mode (4 bits)

CC:0062.01.03.11.005 This field MUST advertise the status of each individual outside door handle.

CC:0062.01.03.11.006 The encoding of the Outside Door Handles Mode bitmask field MUST be according to Table 38 and Table 39.

Table 38, Door Handles Mode bitmask

Bit 3	Bit 2	Bit 1	Bit 0
Handle 4	Handle 3	Handle 2	Handle 1

Table 39, Door Handles Mode bit encoding

Bit value	Description
'0'	Disabled
'1'	Enabled

CC:0062.01.03.11.007 The value '0' MUST signify that the actual handle cannot open the door locally.

CC:0062.01.03.11.008 The value '1' MUST signify that the actual handle can open the door locally.

CC:0062.01.03.13.002 A supporting node MAY advertise just a subset of the available Door Handles Mode values.

CC:0062.01.03.11.009 A controlling node MUST accept all Door Handles Mode values.

Inside Door Handles Mode (4 bits)

CC:0062.01.03.11.00A This field MUST advertise the status of each individual inside door handle.

CC:0062.01.03.11.00B The encoding of the Inside Door Handles Mode bitmask field MUST be according to Table 38 and Table 39.

CC:0062.01.03.13.003 A supporting node MAY advertise just a subset of the available Door Handles Mode values.

CC:0062.01.03.11.00C A controlling node MUST accept all Door Handles Mode values.

Door Condition (8 bits)

CC:0062.01.03.11.00D The Door Condition field MUST advertise the status of the door lock components.

CC:0062.01.03.11.00E The encoding of the Door Condition bitmask field MUST be according to Table 46.

CC:0062.01.03.13.004 A supporting node MAY advertise just a subset of the available Door Condition values.

CC:0062.01.03.11.010 A controlling node MUST accept all Door Condition values.

Lock Timeout Minutes (8 bits)

CC:0062.01.03.11.011 This field MUST advertise the remaining time before the door lock will automatically be locked again.

CC:0062.01.03.11.012 The encoding of the Lock Timeout Minutes field MUST be according to Table 40. The time the supporting node stays unlocked MUST be determined by combining the Lock Timeout Minutes and Lock Timeout Seconds fields.

Table 40, Door Lock Operation Report :: Lock Timeout Minutes

Value	Operation
0x00..0xFD	Unlocked 0 .. 253 minutes (Operation Type = Timed Operation)
0xFE	No unlocked period (Operation Type = Constant Operation)

CC:0062.01.03.11.013 All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

CC:0062.01.03.13.005 A supporting node MAY advertise just a subset of the available Lock Timeout values.

CC:0062.01.03.11.014 A controlling node MUST accept all Lock Timeout values.

Lock Timeout Seconds (8 bits)

CC:0062.01.03.11.015 This field MUST advertise the remaining time before the door lock will automatically be locked again.

CC:0062.01.03.11.016 The encoding of the Lock Timeout Seconds field MUST be according to Table 41. The time to stay unlocked MUST be determined by combining the Lock Timeout Minutes and Lock Timeout Seconds fields.

Table 41, Door Lock Operation Report :: Lock Timeout Seconds

Value	Operation
0..59 (0x00..0x3B)	Unlocked 0 .. 59 seconds (Operation Type = Timed Operation)
254 (0xFE)	No unlocked period (Operation Type = Constant Operation)

CC:0062.01.03.11.017 All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

CC:0062.01.03.13.006 A supporting node MAY advertise just a subset of the available Lock Timeout values.

CC:0062.01.03.11.018 A controlling de MUST accept all Lock Timeout values.

4.37.5 Door Lock Configuration Set Command

This command is used to set the configuration of a supporting door lock device.

CC:0062.01.04.11.001 A door lock device MUST be able to operate with the factory default settings.

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

Operation Type (1 byte)

CC:0062.01.04.11.002 The Operation Type field MUST be set according to Table 42. When timed operation is specified, the Lock Timeout Minutes and Lock Timeout Seconds fields MUST be set to valid values.

Table 42, Door Lock Operation Type

Operation Type	Description	Valid Lock Timeout values
0x01	Constant operation	Minutes = 0xFE Seconds= 0xFE
0x02	Timed operation	Minutes = 0x00..0xFD Seconds= 0x00..0x3B

CC:0062.01.04.11.003 All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

CC:0062.01.04.13.001 A supporting node MAY accept just a subset of the available Operation Type values.

CC:0062.01.04.11.004 A supporting node MUST accept the Operation Type value 0x01.

Outside Door Handles Mode (4 bits)

This field is used to advertise the status of each individual outside door handle.

CC:0062.01.04.11.005 The encoding of the Outside Door Handles Mode bitmask field MUST be according to Table 38 and Table 39.

CC:0062.01.04.13.002 A supporting node MAY ignore the Door Handles Mode value.

Inside Door Handles Mode (4 bits)

CC:0062.01.04.11.006 This field MUST advertise the status of each individual inside door handle.

CC:0062.01.04.11.007 The encoding of the Inside Door Handles Mode bitmask field MUST be according to Table 38 and Table 39.

CC:0062.01.04.13.003 A supporting node MAY ignore the Door Handles Mode value.

Lock Timeout Minutes (1 byte)

CC:0062.01.04.11.008 This field MUST specify the time that a door lock must wait before automatically being locked again.

CC:0062.01.04.11.009 The encoding of the Lock Timeout Minutes field MUST be according to Table 40. The time to stay unlocked MUST be determined by combining the Lock Timeout Minutes and Lock Timeout Seconds fields.

CC:0062.01.04.13.004 A supporting node MAY ignore the Lock Timeout values if it implements only the Operation Type = 0x01 (Constant operation).

Lock Timeout Seconds (1 byte)

CC:0062.01.04.11.00A This field MUST specify the time that a door lock must wait before automatically being locked again.

CC:0062.01.04.11.00B The encoding of the Lock Timeout Seconds field MUST be according to Table 41. The time to stay unlocked MUST be determined by combining the Lock Timeout Minutes and Lock Timeout Seconds fields.

CC:0062.01.04.13.005 A supporting node MAY ignore the Lock Timeout values if it implements only the Operation Type = 0x01 (Constant operation).

4.37.6 Door Lock Configuration Get Command

This command is used to request the configuration parameters of a door lock device.

CC:0062.01.05.11.001 The Door Lock Configuration Report command MUST be returned in response to this command.

CC:0062.01.05.11.002 This command MUST NOT be issued via multicast addressing.

CC:0062.01.05.11.003 A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

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

4.37.7 Door Lock Configuration Report Command

This command is used to advertise the configuration parameters of a door lock device.

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

For fields' description, refer to 4.37.5 Door Lock Configuration Set Command.

4.38 Door Lock Command Class, version 3

The Door Lock Command Class is used to operate and configure a door lock device.

The Door Lock Command Class is an actuator control command class. Refer to 3.6.

Commands and fields not described in version 3 remain unchanged from version 2.

4.38.1 Compatibility considerations

CC:0062.03.00.21.001 A node supporting the Door Lock Command Class, version 3 MUST support the Door Lock Command Class, version 2.

Version 3 adds duration and target value reporting to the Door Lock Operation Report Command.

4.38.2 Door Lock Operation Report Command

This command is used to advertise the status of a door lock device.

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

Fields not described below remain unchanged from previous versions.

Current Door Lock Mode (8 bits)

CC:0062.03.03.11.001 The Current Door Lock Mode field MUST advertise the current mode of the door lock device.

CC:0062.03.03.11.002 The encoding of this field MUST be according to Table 37.

CC:0062.03.03.12.001 An Operation Set Command may be used to initiate a transition to a new target mode. The device may be queried for its current mode while in transition to a new mode. The response to an Operation Get SHOULD be the current mode of the device hardware, e.g. Door/Lock State Unknown. (applies to Version 2 and newer).

Target Door Lock Mode (8 bits)

CC:0062.03.03.11.003 The Target Door Lock Mode field MUST advertise the target mode of an ongoing transition or the most recent transition.

CC:0062.03.03.11.004 The encoding of this field MUST be according to Table 37.

Duration (8 bits)

CC:0062.03.03.12.002 The Duration field SHOULD advertise the remaining time before the target mode is reached.

CC:0062.03.03.11.005 The encoding of this field MUST be according to Table 8.

4.39 Door Lock Command Class, version 4

4.39.1 Terminology

A door lock supports at least 2 modes: secured (it cannot be opened with handles) and unsecured (it can be opened with handles). It may support more modes where the node is for example secured only for the inside handles.

A door lock may have several components such as a latch, a bolt and handles.

A latch, which can also be called a latch bolt or spring bolt, usually locks the door automatically when closing it. The latch bolt is disengaged (retracted) typically when the user turns the door handle, which via the lockset's mechanism, manually retracts the latch bolt, allowing the door to open.

Handles allow to retract the door's latch.

A bolt, also named dead bolt and hook bolt, allows to secure the door. When the bolt is out or extended, the door lock is usually secured (locked) but there are locks that are unlocked even then the bolt is out which will withdraw the bolt when user turns the door handle (which also retract the latch). The position of the bolt is not directly related to the Door Lock Mode.

A supporting door lock runs in constant operation by default. In constant operation, the door lock stays in the same mode until receiving a new command or being locally actuated. A door lock may support timed operation. Timed operation has its own subset of modes, which after being activated will automatically return to the secured mode after the configured Lock Timeout.

An auto-relock functionality can be supported for returning automatically to secured mode some given time after any operation mode change, including constant operation or changes initiated physically by a user (unlocking with key, thumb turn, etc). Auto-relock should not have any effect when the lock runs in Timed Operation

A door lock node may have a twist assist mechanism. When a user starts rotating a key or the thumb turn to unlock the door, a motor starts rotating to help the user completing the movement. It can be helpful for disabled users.

A node may support a hold and release functionality for spring latches. This is useful for locks having a handle that cannot retract the door spring latch on one side. This functionality defines how long the spring latch must be retracted after the door mode has been changed to unsecured.

A block-to-block functionality can be activated for locks having separate lock block on each side of the door and cannot sense local changes on one side. Enabling the block-to-block feature will make the door lock try to execute a lock or unlock command on both sides regardless of the last known lock mode (representing the side that can be sensed). When this feature is disabled, the lock will ignore a command indicating a mode if it is identical to the last known lock mode.

1.1.1 Compatibility considerations

CC:0062.04.00.21.001

The Door Lock Command Class, version 4 is backwards compatible with the Door Lock Command Class, version 3. Fields and commands not mentioned in this version MUST remain unchanged from the Door Lock Command Class, version 3.

Commands for advertising node capabilities are introduced in this version:

- Door Lock Capabilities Get Command
- Door Lock Capabilities Report Command

These commands are extended in order to provide the auto-relock, twist assist and block-to-block functionalities.

- Door Lock Configuration Set Command
- Door Lock Configuration Report Command

4.39.2 Door Lock Operation Set Command

This command is used to set the mode of a supporting door lock node.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_DOOR_LOCK							
Command = DOOR_LOCK_OPERATION_SET (0x01)							
Door Lock Mode							

Door Lock Mode (8 bits)

CC:0062.04.01.11.001 This field is used to specify the operation mode that the supporting node MUST assume. This field MUST be encoded according to Table 43.

Table 43, Door Lock Operation Set::Mode encoding

Value	Mode Description	Operation type	Version
0x00	Door Unsecured	Constant	1
0x01	Door Unsecured with timeout	Timed	1
0x10	Door Unsecured for inside Door Handles	Constant	1
0x11	Door Unsecured for inside Door Handles with timeout	Timed	1
0x20	Door Unsecured for outside Door Handles	Constant	1
0x21	Door Unsecured for outside Door Handles with timeout	Timed	1
0xFF	Door Secured	Constant	1

All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

CC:0062.04.01.51.001 A controlling node MUST NOT specify modes using timeout-based fall back if the Operation Type of the supporting device is set to Constant operation (set by Door Lock Configuration Set).

CC:0062.04.01.11.002 A supporting node MUST ignore modes using timeout-based fall back if Operation Type is set to Constant operation.

CC:0062.04.01.11.003 A supporting node MUST apply the constant modes without any timeout even if it is configured to run in Timed Operation.

CC:0062.04.01.11.004 A supporting node MUST ignore the command if the value specified in this field is not advertised as supported in the Door Lock Capabilities Report Command.

CC:0062.04.01.11.005 A supporting node MUST support the door lock mode values 0x00 and 0xFF.

4.39.3 Door Lock Operation Get Command

This command is used to request the mode of a supporting door lock node.

CC:0062.04.02.11.001 The Door Lock Operation Report Command MUST be returned in response to this command.

CC:0062.04.02.51.001 This command MUST NOT be issued via multicast addressing.

CC:0062.04.02.11.002 A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_DOOR_LOCK							
Command = DOOR_LOCK_OPERATION_GET (0x02)							

4.39.4 Door Lock Operation Report Command

This command is used to advertise the mode of a supporting door lock node.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_DOOR_LOCK							
Command = DOOR_LOCK_OPERATION_REPORT (0x03)							
Current Door Lock Mode							
Outside Door Handles Mode				Inside Door Handles Mode			
Door Condition							
Lock Timeout Minutes							
Lock Timeout Seconds							
Target Door Lock Mode							
Duration							

Current Door Lock Mode (8 bits)

CC:0062.04.03.11.001 This field MUST advertise the current door lock mode of the sending node. This field MUST be encoded according to Table 44.

Table 44, Door Lock Operation Report::Mode encoding

Value	Mode Description	Operation type	Version
0x00	Door Unsecured	Constant (or timed)	1
0x01	Door Unsecured with timeout	Timed	1
0x10	Door Unsecured for inside Door Handles	Constant (or timed)	1
0x11	Door Unsecured for inside Door Handles with timeout	Timed	1
0x20	Door Unsecured for outside Door Handles	Constant (or timed)	1
0x21	Door Unsecured for outside Door Handles with timeout	Timed	1
0xFE	Door mode unknown (e.g. bolt not fully retracted)	N/A	2
0xFF	Door Secured	Constant (or timed)	1

All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

CC:0062.04.03.12.001 A sending node SHOULD set this field to 0xFE if sending this command while performing a transition between 2 modes.

Outside Door Handles Mode (4 bits)

This field is used to advertise the status of each individual outside door handle.

CC:0062.04.03.11.002 This field MUST be treated as a bitmask field and MUST be encoded according to Table 45.

Table 45, Door Lock Operation Report::Door Handles Mode bitmask encoding

Bit 3	Bit 2	Bit 1	Bit 0
Handle 4	Handle 3	Handle 2	Handle 1

CC:0062.04.03.11.003 The value 0 MUST indicate that the actual handle cannot open the door locally.
The value 1 MUST indicate that the actual handle can open the door locally.

Inside Door Handles Mode (4 bits)

This field is used to advertise the status of each individual inside door handle.

CC:0062.04.03.11.004 This field MUST be treated as a bitmask field and MUST be encoded according to Table 45.

CC:0062.04.03.11.005 The value 0 MUST indicate that the actual handle cannot open the door locally.
The value 1 MUST indicate that the actual handle can open the door locally.

Door Condition (8 bits)

This field is used to advertise the state of the door lock components.

CC:0062.04.03.11.006 This field MUST be encoded according to Table 46.

Table 46, Door Lock Operation Report::Door Condition bitmask encoding

Value	Bit 7..3	Bit 2	Bit 1	Bit 0
bit set to '0'	*Reserved*	Latch Open (out, engaged, extended)	Bolt Locked (out, extended)	Door Open
bit set to '1'	*Reserved*	Latch Closed (in, disengaged, retracted)	Bolt Unlocked (in, retracted)	Door Closed

All other bits are reserved and MUST be set to zero by a sending node. Reserved bits MUST be ignored by a receiving node.

CC:0062.04.03.11.007 If a door component is not advertised as supported in the Door Lock Capabilities Report Command, the corresponding bit MUST be set to 0 and MUST be ignored by a controlling node.

Lock Timeout Minutes (8 bits)

This field is used to advertise the remaining time before the door lock mode will automatically be changed to "secured" if operating in timed operation with a timed operation mode.

CC:0062.04.03.11.008 This field MUST be encoded according to Table 47. The remaining time that the sending node will stay in the current mode MUST be determined by combining this field and the *Lock Timeout Seconds* field.

Table 47, Door Lock Operation Report::Lock Timeout Minutes encoding

Value	Description
0..253 (0x00..0xFC)	0..253 minutes (Operation Type and Current mode = Timed Operation)
254 (0xFE)	Undefined

All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

CC:0062.04.03.11.009 A node currently set in a constant operation type mode MUST set this field to 0xFE.

Lock Timeout Seconds (8 bits)

This field is used to advertise the remaining time before the door lock mode will automatically be changed to "secured" if operating in timed operation with a timed operation mode.

CC:0062.04.03.11.00A This field MUST be encoded according to Table 48. The remaining time that the sending node will stay in the current mode MUST be determined by combining this field and the *Lock Timeout Minutes* field.

Table 48, Door Lock Operation Report::Lock Timeout Seconds encoding

Value	Description
0..59 (0x00..0x3B)	0..59 seconds (Operation Type and mode= Timed Operation)
254 (0xFE)	Undefined

All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

CC:0062.04.03.11.00B A node currently set in a constant operation type mode MUST set this field to 0xFE.

Target Door Lock Mode (8 bits)

This field is used to advertise the target mode of an ongoing transition or of the most recent transition.

CC:0062.04.03.11.00C This field MUST be encoded according to Table 44. This field MUST be set to the mode specified in the most-recent received Door Lock Operation Set Command.

CC:0062.04.03.11.00D If the node is going to change mode due to the auto-relock functionality or a timed operation, this field MUST NOT advertise the future mode and MUST be set to the same value as the *Current Door Lock Mode* field.

CC:0062.04.03.11.00E If no transition is ongoing, this field MUST be set to the same value as the *Current Door Lock Mode* field.

CC:0062.04.03.11.00F If the *Current Door Lock Mode* is set to 0xFE, this field MUST indicate the mode is it going to assume when the ongoing transition has been completed.

Duration (8 bits)

CC:0062.04.03.12.002 This field SHOULD advertise the estimated remaining time before the mode indicated in the *Target Door Lock Mode* field is reached.

CC:0062.04.03.11.010 This field MUST be set to 0 if the *Target Door Lock Mode* and *Current Door Lock Mode* fields are set to the same value.

CC:0062.04.03.11.011 The encoding of this field MUST be according to Table 8.

4.39.5 Door Lock Configuration Set Command

This command is used to set the configuration of a supporting door lock node.

CC:0062.04.04.11.001 A supporting node MUST be able to operate with the factory default configuration.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_DOOR_LOCK							
Command = DOOR_LOCK_CONFIGURATION_SET (0x04)							
Operation Type							
Outside Door Handles Mode				Inside Door Handles Mode			
Lock Timeout Minutes							
Lock Timeout Seconds							
Auto-relock time 1 (MSB)							
Auto-relock time 2 (LSB)							
Hold and release time 1 (MSB)							
Hold and release time 2 (LSB)							
Reserved						BTB	TA

Operation Type (1 byte)

CC:0062.04.04.11.002 This field is used to set the operation type at the supporting node. This field MUST be encoded according to Table 49.

Table 49, Door Lock Configuration Set::Operation Type encoding

Value	Description	Version
0x01	Constant operation	1
0x02	Timed operation	1

All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

CC:0062.04.04.11.003 If this field is set to timed operation (0x02), the *Lock Timeout Minutes* and *Lock Timeout Seconds* fields MUST be set to values different than 0xFE.

CC:0062.04.04.11.004 A supporting node MUST ignore this field if it is set to an operation type that is not advertised as supported in the Door Lock Capabilities Report Command.

Outside Door Handles Mode (4 bits)

CC:0062.04.04.11.005 This field is used to enable or disable each individual outside door handle at the supporting node. This field MUST be treated as a bitmask and MUST encoded according to Table 45.

CC:0062.04.04.11.006 The value 0 MUST indicate that the actual handle cannot open the door locally.
The value 1 MUST indicate that the actual handle can open the door locally.

CC:0062.04.04.11.007 A supporting node MUST ignore an outside door handle mode value if it advertises no support for the corresponding handle in the Door Lock Capabilities Report Command.

Inside Door Handles Mode (4 bits)

CC:0062.04.04.11.008 This field is used to enable or disable each individual inside door handle at the supporting node. This field MUST be treated as a bitmask and MUST encoded according to Table 45.

CC:0062.04.04.11.009 The value 0 MUST indicate that the actual handle cannot open the door locally.
The value 1 MUST indicate that the actual handle can open the door locally.

CC:0062.04.04.11.00A A supporting node MUST ignore an inside door handle mode value if it advertises no support for the corresponding handle in the Door Lock Capabilities Report Command.

Lock Timeout Minutes (8 bits)

This field is used to specify the time that the supporting node must wait before returning to the secured mode when receiving timed operation modes in a Door Lock Operation Set Command.

CC:0062.04.04.11.00B Values in the range 0x00..0xFD MUST indicate the actual number of minutes.
The value 0xFE MUST indicate that this field is undefined.

CC:0062.04.04.11.00C This field MUST be set to 0xFE if the *Operation Type* field is set to 0x01 (constant operation)
This field MUST be in the range 0x00..0xFD if the *Operation Type* field is set to 0x02 (timed operation)

CC:0062.04.04.11.00D The time to stay unlocked MUST be determined by combining this field and the *Lock Timeout Seconds* fields.

CC:0062.04.04.11.00E A supporting node MUST ignore this field if it is set to the timed operation type is not advertised as supported in the Door Lock Capabilities Report Command.

Lock Timeout Seconds (8 bits)

This field is used to specify the time that the supporting node must wait before returning to the secured mode when receiving timed operation modes in a Door Lock Operation Set Command.

CC:0062.04.04.11.00F Values in the range 0x00..0x3B MUST indicate the actual number of seconds.
The value 0xFE MUST indicate that this field is undefined.

CC:0062.04.04.11.010 This field MUST be set to 0xFE if the *Operation Type* field is set to 0x01 (constant operation)
This field MUST be in the range 0x00..0x3B if the *Operation Type* field is set to 0x02 (timed operation)

CC:0062.04.04.11.011 The time to stay unlocked MUST be determined by combining this field and the *Lock Timeout Minutes* field.

CC:0062.04.04.11.012 A supporting node MUST ignore this field if it is set to the timed operation type is not advertised as supported in the Door Lock Capabilities Report Command.

Auto-relock time (16 bits)

This field is used to specify the time setting in seconds for auto-relock functionality.

CC:0062.04.04.11.013 This field MUST be ignored by a node advertising no support for the auto-relock functionality in the Door Lock Capabilities Report Command.

CC:0062.04.04.11.014 The value 0 MUST indicate that the auto-relock functionality is disabled and the door lock MUST NOT return to secured mode automatically. (unless using Timed Operation)

CC:0062.04.04.11.015 Values in the range 1..65535 MUST indicate that the auto-relock functionality is enabled and the supporting node MUST return to secured mode after the time in seconds indicated by this field.

Hold and release time (16 bits)

This field is used to specify the time setting in seconds for letting the latch retracted after the supporting node's mode has been changed to unsecured.

CC:0062.04.04.11.016 This field MUST be ignored by a node advertising no support for the hold and release functionality in the Door Lock Capabilities Report Command.

CC:0062.04.04.11.017 The value 0 MUST indicate that the hold and release functionality is disabled and the supporting node MUST NOT keep the latch retracted when the door lock mode becomes unsecured.

CC:0062.04.04.11.018 Values in the range 1..65535 MUST indicate that the hold and release functionality is enabled and the door lock latch MUST keep open according to the time in seconds indicated by this field.

TA (Twist assist) (1 bit)

CC:0062.04.04.11.019 This field MUST indicate if the twist assist functionality is enabled.

CC:0062.04.04.11.01A This field MUST be ignored by a node advertising no support for the twist assist functionality in the Door Lock Capabilities Report Command.

CC:0062.04.04.11.01B The value 0 MUST indicate that the twist assist functionality MUST be disabled at the supporting node. The value 1 MUST indicate that the twist assist functionality MUST be enabled at the supporting node.

BTB (Block-to-block) (1 bit)

This field is used to indicate if the block-to-block functionality is enabled.

CC:0062.04.04.11.01C This field MUST be ignored by a node advertising no support for the block-to-block functionality in the Door Lock Capabilities Report Command.

CC:0062.04.04.11.01D The value 0 MUST indicate that the block-to-block functionality is disabled and the supporting node MAY ignore Door Lock Operation Set Commands if it detects to be already in the specified mode.

CC:0062.04.04.11.01E The value 1 MUST indicate that the block-to-block functionality is enabled and the supporting node MUST activate its motors until blocked to try to reach the mode indicated by Door Lock Operation Set Commands, even if it detects to be already in the specified mode.

4.39.6 Door Lock Configuration Get Command

This command is used to request the configuration parameters of a door lock device.

CC:0062.04.05.11.001 The Door Lock Configuration Report Command MUST be returned in response to this command.

CC:0062.04.05.51.001 This command MUST NOT be issued via multicast addressing.

CC:0062.04.05.11.002 A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_DOOR_LOCK							
Command = DOOR_LOCK_CONFIGURATION_GET (0x05)							

4.39.7 Door Lock Configuration Report Command

This command is used to advertise the configuration parameters of a door lock device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_DOOR_LOCK							
Command = DOOR_LOCK_CONFIGURATION_REPORT (0x06)							
Operation Type							
Outside Door Handles Mode				Inside Door Handles Mode			
Lock Timeout Minutes							
Lock Timeout Seconds							
Auto-relock time 1 (MSB)							
Auto-relock time 2 (LSB)							
Hold and release time 1 (MSB)							
Hold and release time 2 (LSB)							
Reserved						BTB	TA

CC:0062.04.06.11.001 Fields in this command MUST be according to descriptions in 4.39.5 Door Lock Configuration Set Command

1.1.2 Door Lock Capabilities Get Command

This command is used to request the Door Lock capabilities of a supporting node.

CC:0062.04.07.11.001 The Door Lock Capabilities Report Command MUST be returned in response to this command.

CC:0062.04.07.51.001 This command MUST NOT be issued via multicast addressing.

CC:0062.04.07.11.002 A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_DOOR_LOCK							
Command = DOOR_LOCK_CAPABILITIES_GET (0x07)							

1.1.3 Door Lock Capabilities Report Command

This command is used to advertise the Door Lock capabilities supported by the sending node.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_DOOR_LOCK							
Command = DOOR_LOCK_CAPABILITIES_REPORT (0x08)							
Reserved				Supported Operation type Bit Mask Length			
Supported Operation Type Bit Mask 1							
...							
Supported Operation Type Bit Mask N							
Supported Door Lock Mode List Length							
Supported Door Lock Mode 1							
…							
Supported Door Lock Mode M							
Supported Outside Handle Modes Bitmask				Supported Inside Handle Modes Bitmask			
Supported door components							
Reserved				ARS	HRS	TAS	BTBS

Reserved

CC:0062.04.08.11.001 This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Supported Operation Type Bit Mask Length (5 bits)

This field is used to advertise the length in bytes of the *Supported Operation Type Bit Mask* field carried in the command.

CC:0062.04.08.11.002 This field MUST be set to the minimum value which allows to advertise all supported Operation types.

Supported Operation Type Bit Mask (N bytes)

CC:0062.04.08.11.003 This field is used to advertise the supported Door Lock Operation Types values. The length of this field in bytes MUST match the value advertised in the *Supported Operation Type Bit Mask Length* field.

CC:0062.04.08.11.004 This field MUST be encoded as follows:

- Bit 0 in Bit Mask 1 represents Operation Type 0x00 (Reserved)
- Bit 1 in Bit Mask 1 represents Operation Type 0x01 (Constant operation)
- Bit 2 in Bit Mask 1 represents Operation Type 0x02 (Timed operation)
- …

Valid Operation Type values are defined in Table 49.

CC:0062.04.08.11.005 If an Operation Type is supported, the corresponding bit MUST be set to '1'.
If an Operation Type is not supported, the corresponding bit MUST be set to '0'.

CC:0062.04.08.11.006 All other bits are reserved and MUST be set to zero by a sending node. Reserved bits MUST be ignored by a receiving node.

CC:0062.04.08.11.007 A node MUST support the constant operation type 0x01.

Supported Door Lock Mode List Length (8 bits)

CC:0062.04.08.11.008 This field MUST advertise the length in bytes of the *Supported Door Lock Mode* field carried in the command.

Supported Door Lock Mode (M bytes)

CC:0062.04.08.11.009 This field is used to advertise the supported Door Lock Modes values. The length of this field in bytes MUST match the value advertised in the *Supported Door Lock Mode List Length* field.

CC:0062.04.08.11.00A Each byte MUST indicate the value of a supported Door Lock Mode that can be specified in a Door Lock Operation Set Command. Each byte MUST be encoded according to Table 43.

CC:0062.04.08.11.00B A sending node MUST advertise support for the Door Lock modes 0x00 and 0xFF.

Supported Outside Handle Modes Bitmask (4 bits)

This field is used to advertise which outside handles can be enabled and disabled using the Door Lock Configuration Set Command.

CC:0062.04.08.11.00C This field MUST be treated as a bitmask and represent outside door handles as described in Table 45.

CC:0062.04.08.11.00D If the corresponding handle can be enabled and disabled, the corresponding bit MUST be set to 1.
If the corresponding handle cannot be enabled or disabled, the corresponding bit MUST be set to 0.

Supported Inside Handle Modes Bitmask (4 bits)

This field is used to advertise which inside handles can be enabled and disabled using the Door Lock Configuration Set Command.

CC:0062.04.08.11.00E This field MUST be treated as a bitmask and represent inside door handles as described in Table 45.

CC:0062.04.08.11.00F If the corresponding handle can be enabled, the corresponding bit MUST be set to 1.
If the corresponding handle cannot be enabled, the corresponding bit MUST be set to 0.

Supported Door components (8 bits)

CC:0062.04.08.11.010 This field MUST advertise the supported door lock component that can be reported in the Door Lock condition field of the Door Lock Operation Report Command.

CC:0062.04.08.11.011 This field MUST be treated as a bitmask and represent inside door handles as described in Table 50

Table 50, Door Lock Capabilities Report::Supported Door components encoding

Bit 7..3	Bit 2	Bit 1	Bit 0
Reserved	Latch	Bolt	Door

CC:0062.04.08.11.012 If the corresponding component is supported, the corresponding bit MUST be set to ‘1’.
If the corresponding component is not supported, the corresponding bit MUST be set to ‘0’.

ARS (Auto-Relock support) (1 bit)

This field is used to advertise if the sending node supports the auto-relock functionality.

CC:0062.04.08.11.013 The value 1 MUST indicate that the auto-relock functionality is supported.
The value 0 MUST indicate that the auto-relock functionality is not supported.

HRS (Hold and release support) (1 bit)

CC:0062.04.08.11.014 This field MUST advertise if the hold and release functionality is supported.

CC:0062.04.08.11.015 The value 0 MUST indicate that the hold and release functionality is not supported
The value 1 MUST indicate that the hold and release functionality is supported.

TAS (Twist assist support) (1 bit)

CC:0062.04.08.11.016 This field MUST advertise if the Twist Assist functionality is supported.

CC:0062.04.08.11.017 The value 0 MUST indicate that the twist assist functionality is not supported
The value 1 MUST indicate that the twist assist functionality is supported.

BTBS (Block-to-block support) (1 bit)

This field is used to advertise if the block-to-block functionality is supported.

CC:0062.04.08.11.018 The value 0 MUST indicate that the block-to-block functionality is not supported
The value 1 MUST indicate that the block-to-block functionality is supported.

4.40 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.

4.40.1 Door Lock Logging Records Supported Get Command

This command is used to request the number of records that the audit trail supports.

The Door Lock Logging Records Supported Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

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

4.40.2 Door Lock Logging Records Supported Report Command

This command is used to report the maximum number of reports 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_REPORT							
Max records stored							

Max records stored (8 bits)

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

4.40.3 Door Lock Logging Record Get Command

This command is used to request the audit trail.

The Door Lock Logging Record Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

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

Record number (8 bits)

The Record number field indicates the record to be requested.

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.

4.40.4 Door Lock Logging Record Report Command

This command returns records from the audit trail.

To provide flexibility the user associated with the record may be identified by one of two methods: the user identifier or the user code.

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_CODE 1							
..							
USER_CODE N							

Record number (8 bits)

Record number requested (1- 255).

Timestamp - Year (16 bits)

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

Timestamp - Month (8 bits)

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

Timestamp - Day (8 bits)

Specify the day of the month between 01 and 31.

Record status (3-bits)

The Record Status field is used to indicate if 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 bits)

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

Timestamp - 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.

Timestamp - 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.

Note: If RTC (Real Time Clock) all values in Time Stamp record SHOULD be set to 0. Most recent Record MUST be stored under Record Number 1.

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 bits)

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_CODE (N bytes)

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.

Event Type (8 bits)

This field MUST be encoded according to Table 51.

Table 51, Door Lock Logging Record Report::Event Type encoding

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 ones are to be supported.

4.41 Energy Production Command Class, version 1

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

4.41.1 Energy Production Get Command

This command is used to request various production data from the device.

The Energy Production Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

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

All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

4.41.2 Energy Production Report Command

This command is used to retrieve various production data from the device.

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 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 (N bytes)

The value is MUST be treated as a signed field (Table 10). The field MAY be 1, 2 or 4 bytes in size. The first byte is the most significant byte.

4.42 Entry Control Command Class, version 1

The Entry Control Command Class defines a method for advertising user input to a central Entry Control application and for the discovery of capabilities. User input may be button presses, RFID tags or other means.

It is RECOMMENDED that this command class is only supported via secure communication. This recommendation may be raised to a stronger requirement at the Device Class or Device Type level.

The Entry Control Command Class provides the following commands:

- Key Supported Get/Report – Used by the controller to request which ASCII keys are present on the Entry Control device.
- Event Supported Get/Report – Used by the controller to request which Event Types are supported by the Entry Control device.
- Configuration Set/Get/Report – Used by the controller to configure the Entry Control device
- Notification Command – Used by the Entry Control device to notify the controller of inputs.

4.42.1 Interoperability Considerations

It is RECOMMENDED that an Entry Control input device that implements the Entry Control Command Class also implements the Indicator Command Class, version 2.

If implementing the Indicator Command Class, version 2, it is RECOMMENDED that all indicator resources are addressable via the Indicator Command Class, version 2.

The input device MAY implement local control of indicator resources. If the Indicator Command Class, version 2 is supported, the first received Indicator Set command MUST disable all local control of indicator resources – except for interface feedback functionality like a short beep on each button press.

Transmissions may fail due to central control application fault or due to RF jamming. It is RECOMMENDED that an Entry Control input device provides local user feedback if the transmission of notifications to the central control application fails.

If the device only features light indicators, it is RECOMMENDED that all light flashes at 2Hz for at least 5 seconds to indicate transmission error.

If the device features a buzzer, it is RECOMMENDED that the buzzer generates sound pulses at a rate of 2Hz for at least 5 seconds to indicate transmission error.

If implementing the indicator Command Class, version 2, in devices based on Role Type RSS (using Wakeup Command Class), the control of indicators must be synchronized with the Wakeup Notification. The controller must therefore wait for the Wakeup Notification, before it can control the indictors. The device implementing the indicator Command Class must therefore send the Wakeup Notification after sending the Entry Control Notification Command, in case the device expects to be controlled.

4.42.2 Security Considerations

An attacker could theoretically determine the length of manually entered user credentials even if they are transmitted via encryption. It is therefore MANDATORY to add padding bytes to ASCII strings so that all transferred ASCII strings are structured as one or more blocks of 16 characters. The ASCII code 0xFF MUST be reserved for padding purposes and it MUST NOT be used for any other purposes.

4.42.3 Handling user supplied data

The Entry Control device must cache the user input before sending the full entry in one Notification Frame. A user input MUST therefore have a termination, which MAY be determined by:

- The Key Cache Size is exceeded
- The Key Cache Timeout is exceeded
- A Command Button is pressed
- User data is received by other means, e.g. from an RFID tag

Key Cached Size:

The Key Cached Size is configured to specify the number of user inputs before the Notification Frame is sent. After sending the Notification Frame the cache must be cleared and subsequent user inputs MUST be considered a new entry.

The Key Cached Size May be configured to 1, in which case a Notification Frame is send for each user input.

It is RECOMMENDED to have a default Key Cached Size of 4, in which case the Notification Frame will be send after 4 entries.

Key Timeout:

The Key Timeout is configured to specify the maximum time between user inputs. If the time between user inputs exceeds the Timeout, the cached user inputs will be send in a Notification Frame. After sending the Notification Frame the cache must be cleared and subsequent user inputs MUST be considered a new entry.

Based on Command Button:

A user input may be terminated by the user pressing on of the Command Buttons like ENTER or ARM_ALL. The cached entry will be sent in a Notification Frame immediately after the user presses the Command Button. After sending the Notification Frame the cache must be cleared and subsequent user inputs MUST be considered a new entry.

Based on RFID:

When presenting an RFID tag, the ID will be read from the tag, and this terminates the user input. So the cached ID ‘s from the RFID tag data can immediately be sent in a Notification Frame. After sending the Notification Frame the cache must be cleared and subsequent user inputs MUST be considered a new entry.

Presenting an RFID tag may also terminate an ongoing user input. For instance, a user may enter four characters and subsequently present an RFID tag. In that case, the four entries must first be sent in one Notification Frame, and subsequently the ID ‘s from the RFID tag data is sent in a second Notification Frame.

4.42.4 Handling Incorrect Entry

A user may do an incorrect entry e.g. the credential is 1234 but the user enters 1734. In this case the terminal may provide a “delete” option, to allow the user to fix or re-enter the credential. If a “delete” option is not provided, the invalid code must first be send, followed by a new (correct) entry.

The terminal must not assume that the receiving controller does error handling like 1+7+Delete+2+3+4 = 1234.

4.42.5 Entry Control Notification Command

This command is used to advertise user input.

Depending on the Event Type, this command MAY carry manually entered user credentials.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ENTRY_CONTROL							
Command = ENTRY_CONTROL_NOTIFICATION							
Sequence Number							
Reserved						Data Type	
Event Type							
Event Data Length							
Event Data 1 (optional)							
...							
Event Data N (optional)							

Sequence Number (8 bit)

The sequence number MUST be incremented each time a new command of this type is issued.

The value MUST be in the range 0..255. The initial value MAY be any value in the range 0..255. A receiving device MUST use the Sequence Number to detect and ignore duplicates.

Reserved

This field MUST be set to 0 by a sending device and MUST be ignored by a receiving device.

Data Type (2 bits)

This field is used to advertise the type of data (if any) that is appended to this command.

Table 52, Entry Control Notification :: Event Data Type

Data Type		Description
0x00	NA	No data included
0x01	RAW	1 to 32 bytes of arbitrary binary data
0x02	ASCII	1 to 32 ASCII encoded characters. ASCII codes MUST be in the value range 0x00-0xF7. The string MUST be padded with the value 0xFF to fit 16 byte blocks when sent in a notification.
0x03	MD5	16 byte binary data encoded as a MD5 hash value.

All other values are reserved and MUST NOT be used by a sending device. Reserved values MUST be ignored by a receiving device.

Event Type (4 bit)

This field is used to advertise the actual Event Type.

The field MUST be encoded according to section 4.42.13

Event Data Length (8 bit)

This field MUST advertise the length of the Event Data field in bytes.

The value MUST be in the range 0..32.
If no data bytes are included, this field MUST be set to 0.

Event Data (n bytes)

This field is used to carry data related to the event, e.g. received from an RFID tag.

The length of this field MUST comply with the length advertised by the Event Data Length field.
The format of this field MUST comply with the data format advertised by the Data Type field.

4.42.6 Entry Control Key Supported Get Command

This command is used to query the keys that a device implements for entry of user credentials.

The Entry Control Key Supported Report command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ENTRY_CONTROL							
Command = ENTRY_CONTROL_KEY_SUPPORTED_GET							

4.42.7 Entry Control Key Supported Report Command

This command is used to advertise the keys that a device implements for entry of user credentials.

A management interface may determine the available keys for credential entry from this command.
The range of available Command Keys may be determined via the Entry Control Event Supported Report Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ENTRY_CONTROL							
Command = ENTRY_CONTROL_KEY_SUPPORTED_REPORT							
Key Supported Bit Mask Length							
Key Supported Bit Mask 1							
…							
Key Supported Bit Mask N							

Key Supported Bit Mask Length

This field is used to advertise the number of Key Supported Bit Mask Bytes to follow.

Only the Key Supported Bit Mask bytes until the last supported ASCII key SHOULD be included.

Key Supported Bit Mask (Variable length)

This field is used to advertise the keys that a device implements for entry of user credentials.

The Key Supported Bit Mask field MUST advertise ASCII codes that represent the supported keys.

- Bit 0 in Bit Mask 1 indicates that the supporting device may issue ASCII code 0
- Bit 1 in Bit Mask 1 indicates that the supporting device may issue ASCII code 1
- …
- Bit 7 in Bit Mask 16 indicates that the supporting device may issue ASCII code 127

4.42.8 Entry Control Event Supported Get Command

This command is used to request the supported Events of a device.

The Entry Control Event Supported Report command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ENTRY_CONTROL							
Command = ENTRY_CONTROL_EVENT_SUPPORTED_GET							

4.42.9 Entry Control Event Supported Report Command

This command is used to advertise the supported Event Types.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ENTRY_CONTROL							
Command = ENTRY_CONTROL_EVENT_SUPPORTED_REPORT							
Reserved						Data Type Supported Bit Mask Length	
Data Type Supported Bit Mask 1							
…							
Data Type Supported Bit Mask M							
Reserved				Event Supported Bit Mask Length			
Event Type Supported Bit Mask 1							
…							
Event Type Supported Bit Mask N							
Key Cached Size supported Minimum							
Key Cached Size supported Maximum							
Key Cached Timeout supported Minimum							
Key Cached Timeout supported Maximum							

Data Type Supported Bit Mask Length (2 bits)

This field is used to advertise the number of Data Type Supported Bit Mask Bytes to follow.

Reserved

This field MUST be set to 0 by a sending device and MUST be ignored by a receiving device.

Data Type Supported Bit Mask (Variable length)

This field is used to advertise the supported Data Types.

- Bit 0 in Bit Mask 1 indicates if Data Type 0 is supported
- Bit 1 in Bit Mask 1 indicates if Data Type 1 is supported
- …

For the definition of Data Type IDs, refer to Table 52.

Event Type Supported Bit Mask Length (1 byte)

This field is used to advertise the number of Event Type Supported Bit Mask Bytes to follow.

Event Type Supported Bit Mask (variable length)

This field is used to advertise the supported Event Type.

- Bit 0 in Bit Mask 1 indicates if Event Type 0 is supported
- Bit 1 in Bit Mask 1 indicates if Event Type 1 is supported
- …

For the definition of Event Type IDs, refer to Table 53.

Key Cached Size Supported Minimum

The minimum configurable number of key entries before the CACHED_KEYS notification is sent.
Refer to 4.42.10

Key Cached Size Supported Maximum

The maximum configurable number of key entries before the CACHED_KEYS notification is sent.
Refer to 4.42.10

Key Cached Timeout Supported Minimum

The minimum configurable timeout before the CACHED_KEYS notification is sent.
Refer to 4.42.10

Key Cached Timeout Supported Maximum

The maximum configurable timeout before the CACHED_KEYS notification is sent.
Refer to 4.42.10

4.42.10 Entry Control Configuration Set Command

This command is used to configure Event Type specific parameters.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ENTRY_CONTROL							
Command = ENTRY_CONTROL_CONFIGURATION_SET							
Key Cache Size							
Key Cache Timeout							

Key Cache Size (1 byte)

This field specifies the number of characters the key cache MUST store before sending data to the central control application. Data MUST be sent when the number of characters in the cache matches or exceeds the value of this field.

The value MUST be in the range 1..32. The default value of this field SHOULD be 4.

In deployments where characters are entered first and a command button is pressed subsequently, it is RECOMMENDED that this field is set to 32 and that the Key Cache Timeout is set to 2 seconds.

Key Cache Timeout (1 byte)

This field specifies the number of seconds the key cache MUST wait for additional characters before sending data to the central control application. Data MUST be sent if a Key Cache Timeout occurs.

The Key Cache Timeout MUST be measured from the most recent reception of a character.

The value SHOULD be in the range 1..10 seconds. The default value SHOULD be 2 seconds.

4.42.11 Entry Control Configuration Get Command

This command is used to request the operational mode of a device.

The Entry Control Configuration Report command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ENTRY_CONTROL							
Command = ENTRY_CONTROL_CONFIGURATION_GET							

4.42.12 Entry Control Configuration Report Command

This command is used to advertise the current operational mode of a device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ENTRY_CONTROL							
Command = ENTRY_CONTROL_CONFIGURATION_REPORT							
Key Cache Size							
Key Cache Timeout							

Key Cache Size (1 byte)

Refer to 4.42.10 Entry Control Configuration Set Command.

Key Cache Timeout (1 byte)

Refer to 4.42.10 Entry Control Configuration Set Command.

4.42.13 Event Types

The Event Types MUST be encoded according to Table 53.

Table 53, Event Type and Event identifiers

Event Type		Recommended Button Label	Event Data (optional)
0x00	CACHING	-	-
0x01	CACHED_KEYS	-	ASCII bytes
0x02	ENTER	Enter	ASCII bytes
0x03	DISARM_ALL	Disarm	ASCII bytes
0x04	ARM_ALL	Arm	ASCII bytes
0x05	ARM_AWAY	Away	ASCII bytes
0x06	ARM_HOME	Home	ASCII bytes
0x07	EXIT_DELAY	Arm_Delay	ASCII bytes
0x08	ARM_1	Arm zone 1	ASCII bytes
0x09	ARM_2	Arm zone 2	ASCII bytes
0x0A	ARM_3	Arm zone 3	ASCII bytes
0x0B	ARM_4	Arm zone 4	ASCII bytes
0x0C	ARM_5	Arm zone 5	ASCII bytes
0x0D	ARM_6	Arm zone 6	ASCII bytes
0x0E	RFID	-	As advertised by the Data Type field
0x0F	BELL	🔔	-
0x10	FIRE	Fire	-
0x11	POLICE	Police	-
0x12	ALERT_PANIC		-
0x13	ALERT_MEDICAL		-
0x14	GATE_OPEN	Open, Up. ’O’ or similar	ASCII bytes
0x15	GATE_CLOSE	Close, Down, ’C’ or similar	ASCII bytes
0x16	LOCK	🔒	ASCII bytes
0x17	UNLOCK	🔓	ASCII bytes
0x18	TEST	Test	ASCII bytes

0x19	CANCEL	Cancel	ASCII bytes

All other values are reserved and MUST NOT be used by a sending device. Reserved values MUST be ignored by a receiving device.

Event Type CACHING

The CACHING Event Type is used to indicate to the central controller that the user has started entering credentials, and that caching is initiated. This allows the central controller to change the indications on the Entry Control device through the indicator Command Class, or to change the status of the central controller user interface.

If Key Cached Size it set to 1, the CACHING event notification MUST NOT be sent. Instead each individual credential byte MUST be sent in its own CACHED_KEYS event notification.

Event Type CACHED_KEYS

The CACHED_KEYS Event Type is used to send user inputs in a Notification Frame. The CACHED_KEYS is sent when the user input is terminated by one of the following reasons:

- The Key Cache Size is exceeded
- The Key Cache Timeout is exceeded
- A command button is pressed
- User data is received by other means, e.g. from an RFID tag

4.43 Generic Schedule Command Class, version 1

The Generic Schedule Command Class is used to define schedules.

Schedules defined in this command class are essentially time ranges or intervals. Each schedule is associated to a Schedule ID.

The Schedules defined in this Command Class can be used in other Command Classes, e.g. to schedule when user credentials are valid or when to apply a particular configuration.

4.43.1 Terminology

Schedules are composed by one or more Time Range(s). A time range is a composed of one or several scheduling parameters, such as weekdays, day of the month, hour of the day, etc.

Time ranges are defined individually with scheduling parameters. For example, it could be Time Range 1 representing weekdays during work hours and Time Range 2 representing public holidays at fixed dates. (e.g. 25th of December and 31st of October)

A schedule can then be set including or excluding each of its composing time ranges. Schedule ID 1 could include weekdays work hours (Time Range 1) and exclude public holidays (Time Range 2). It means that Schedule 1 must be active only on weekdays work hours when it is not a public holiday.

At least one of the included Time Ranges conditions must be fulfilled and none of the excluded Time Ranges must be fulfilled for a Schedule to be active.

4.43.2 Interoperability considerations

This Command Class relies on time and date synchronization between controlling and supporting nodes.

Nodes supporting this command class should also have correct time or date settings depending on their capabilities. Supporting nodes may support the Clock Command Class for this purpose. The Time Command Class may also be used for reading the current date and time at the supporting node.

1.1.1 Generic Schedule Capabilities Get Command

This command is used to request the scheduling capabilities of a supporting node.

The Generic Schedule Capabilities Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_GENERIC_SCHEDULE							
Command = GENERIC_SCHEDULE_CAPABILITIES_GET							

1.1.2 Generic Schedule Capabilities Report Command

This command is used to advertise the number of supported Schedule Slot IDs for each Schedule Type/User Identifier by the sending node.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_GENERIC_SCHEDULE							
Command = GENERIC_SCHEDULE_CAPABILITIES_REPORT							
Number of supported Schedule IDs (MSB)							
Number of supported Schedule IDs (LSB)							
Res	Number of supported Time Range IDs (MSB)						
Number of supported Time Range IDs (LSB)							
Number of supported Time Ranges per Schedule							
Reserved					Week-days	Date	Hour & Minute

Number of supported Schedule IDs (16 bits)

This field is used to indicate how many Schedule IDs are supported by a supporting node.

This field MUST be in the range 1..65535.

Number of supported Time Range IDs (15 bits)

This field is used to indicate how many Time Ranges IDs are supported by a supporting node.

This field MUST be in the range 1.. 32767.

The number of supported Time Ranges IDs should be equal or greater than the (*Number of supported Schedule IDs*) x (*Number of supported Time Ranges per Schedule*)

Number of supported Time Ranges per Schedule (8 bits)

This field is used to indicate how many time ranges can be used as part of 1 schedule.

This field MUST be in the range 1..255.

Weekdays (1 bit)

This field is used to indicate if the supporting node can accept weekdays in the time ranges.

The value 0 MUST indicate that weekdays are not supported in time ranges and will be ignored.
The value 1 MUST indicate that weekdays are supported in time ranges.

Date (1 bit)

This field is used to indicate if the supporting node can accept years, month and day parameters in the time ranges.

The value 0 MUST indicate that start/stop dates are not supported in time ranges and will be ignored.
The value 1 MUST indicate that start/stop dates are supported in time ranges.

Hour & Minute (1 bit)

This field is used to indicate if the supporting node can accept start and stop hours/minutes in the time ranges.

The value 0 MUST indicate that start/stop hours/minutes are not supported in time ranges and will be ignored.
The value 1 MUST indicate that start/stop hours/minutes are supported in time ranges.

A supporting node MUST advertise support for at least 1 scheduling parameter (Weekdays, Date, Hour & Minute).

1.1.3 Generic Schedule Time Range Set Command

This command is used to configure a Time Range for a supporting node.

A Time Range MUST be active when all the conditions indicated in its configuration are fulfilled. Some examples are provided in 4.43.2.1.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_GENERIC_SCHEDULE							
Command = GENERIC_SCHEDULE_TIME_RANGE_SET							
Res	Time Range ID 1 (MSB)						
Time Range ID 2 (LSB)							
In use	Weekday bitmask						
In use	Start Year (MSB)						
Start Year (LSB)							
In use	Stop Year (MSB)						
Stop Year (LSB)							
In use	Reserved			Start Month			
In use	Reserved			Stop Month			
In use	Reserved			Start Day			
In use	Reserved			Stop Day			
In use	Reserved			Start Hour			
In use	Reserved			Stop Hour			
In use	Res		Start Minute				
In use	Res		Stop Minute				
In use	Reserved			Daily Start Hour			
In use	Reserved			Daily Stop Hour			
In use	Res		Daily Start Minute				
In use	Res		Daily Stop Minute				

Res / Reserved

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Time Range ID (15 bits)

This field is used to indicate which Time Range ID to use for the Time Range being set.

This field MUST be in the range 0..[Number of supported Time Range IDs].

The value 0 MUST indicate that the supporting node MUST erase all configured Time Ranges. In this case, all other fields MUST be set to 0 and ignored by a receiving node.
Values in the range 1..[Number of supported Time Range IDs] MUST indicate the actual Time Range ID to configure.

If a non-supported Time Range ID is specified in this command, a receiving node MUST ignore the command.

In use (15 x 1 bit)

This field is used to indicate if the corresponding scheduling parameter must be used in the Time Range being set.

This field MUST apply to the scheduling parameter being part of the same byte. E.g. the “in use” field part of the byte comprising the *Weekday bitmask* field MUST indicate if the *Weekday bitmask* parameter MUST be used and applied to the current Time Range.

The value 0 MUST indicate that the corresponding field is set to 0 and MUST be ignored.
The value 1 MUST indicate that the corresponding field is set to a valid value and MUST be applied for the actual Time Range ID.

All “in use” fields set to 0 MUST indicate to erase a Time Range configuration.

Weekday bitmask (7 bits)

This field is used to indicate on which days the actual Time Range ID MUST be active.

This field MUST be treated as a bitmask and encoded according to Table 54.

Table 54, Generic Schedule Set::Weekday bitmask encoding

Bit #	Weekday
Bit 0	Sunday
Bit 1	Monday
Bit 2	Tuesday
Bit 3	Wednesday
Bit 4	Thursday
Bit 5	Friday
Bit 6	Saturday

If the corresponding *In use* field is set to 1:

- The value 0 MUST indicate that the Time Range MUST NOT be active during the corresponding weekday.
- The value 1 MUST indicate that the Time Range MUST be active during the corresponding weekday.

If the corresponding *In use* field is set to 0, the Time Range MUST be active regardless of the current weekday.

A supporting node MUST ignore this field if it advertises no support for the weekday scheduling parameter in the Generic Schedule Capabilities Report Command

Start Year (15 bits)

This field is used to indicate the start year for the actual Time Range ID.

If the corresponding *In use* field is set to 1, the fields MUST indicate that the Time Range ID MUST be use this field as parameter. Refer to 4.43.2.1 for details. This field MUST be encoded using unsigned representation.

If the corresponding *In use* field is set to 0, the Time Range MUST ignore this parameter.

A supporting node MUST ignore this field if it advertises no support for the Date scheduling parameters in the Generic Schedule Capabilities Report Command

Stop Year (15 bits)

This field is used to indicate the stop year for the actual Time Range ID.

If the corresponding *In use* field is set to 1, the fields MUST indicate that the Time Range ID MUST be use this field as parameter. Refer to 4.43.2.1 for details. This field MUST be encoded using unsigned representation.

If the corresponding *In use* field is set to 0, the Time Range MUST ignore this parameter.

A supporting node MUST ignore this field if it advertises no support for the Date scheduling parameters in the Generic Schedule Capabilities Report Command

Start Month (4 bits)

This field is used to indicate the start month for the actual Time Range ID.

If the corresponding *In use* field is set to 1, the fields MUST indicate that the Time Range ID MUST be use this field as parameter. Refer to 4.43.2.1 for details. The values 1..12 MUST represent respectively January..December.

If the corresponding *In use* field is set to 0, the Time Range MUST ignore this parameter.

A supporting node MUST ignore this field if it advertises no support for the Date scheduling parameters in the Generic Schedule Capabilities Report Command

Stop Month (4 bits)

This field is used to indicate the stop month for the actual Time Range ID.

If the corresponding *In use* field is set to 1, the fields MUST indicate that the Time Range ID MUST be use this field as parameter. Refer to 4.43.2.1 for details. The values 1..12 MUST represent respectively January..December.

If the corresponding *In use* field is set to 0, the Time Range MUST ignore this parameter.

A supporting node MUST ignore this field if it advertises no support for the Date scheduling parameters in the Generic Schedule Capabilities Report Command

Start Day (5 bits)

This field is used to indicate the start day of the month for the actual Time Range ID.

If the corresponding *In use* field is set to 1, the fields MUST indicate that the Time Range ID MUST be use this field as parameter. Refer to 4.43.2.1 for details. The field MUST be in the range 1..31.

If the corresponding *In use* field is set to 0, the Time Range MUST ignore this parameter.

A supporting node MUST ignore this field if it advertises no support for the Date scheduling parameters in the Generic Schedule Capabilities Report Command

Stop Day (5 bits)

This field is used to indicate the stop day of the month for the actual Time Range ID.

If the corresponding *In use* field is set to 1, the fields MUST indicate that the Time Range ID MUST be use this field as parameter. Refer to 4.43.2.1 for details. The field MUST be in the range 1..31.

If the corresponding *In use* field is set to 0, the Time Range MUST ignore this parameter.

A supporting node MUST ignore this field if it advertises no support for the Date scheduling parameters in the Generic Schedule Capabilities Report Command

Start Hour /Stop Hour (5 bits) & Start Minute / Stop Minute (6 bits)

These fields are used to indicate from and until which times the actual Time Range ID MUST be active.

If the corresponding *In use* field is set to 1:

- The *Start Hour / Stop Hour* fields MUST be in the range 0..23.
- The *Start Minute / Stop Minute* fields MUST be in the range 0..59.

If the corresponding *In use* field is set to 0, the corresponding field MUST be ignored by the Time Range.

A supporting node MUST ignore this field if it advertises no support for the Hour & Minute scheduling parameter in the Generic Schedule Capabilities Report Command

Daily Start Hour /Stop Hour (5 bits) & Daily Start Minute / Stop Minute (6 bits)

These fields are used to indicate in which time range the actual Time Range ID MUST be active.

If the corresponding *In use* field is set to 1:

- The fields MUST indicate that the Time Range ID MUST be active daily between the 2 specified hour and minutes times. (e.g. from the 08:00 to 16:30)
- The *Start Hour / Stop Hour* fields MUST be in the range 0..23.
- The *Start Minute / Stop Minute* fields MUST be in the range 0..59.

If the corresponding *In use* field is set to 0, the corresponding field MUST be ignored by the Time Range.

A supporting node MUST ignore this field if it advertises no support for the Hour & Minute scheduling parameter in the Generic Schedule Capabilities Report Command

4.43.2.1 Time Range parameters and rules

Time Ranges MUST be active when all their scheduling parameters are matching the conditions. We define the parameter groups shown in Table 55.

Table 55, Generic Schedule Set Command::Parameter groups

Parameter group	Fields
Daily Start time	Daily Start Hour Daily Start Minute
Daily Stop time	Daily Stop Hour Daily Stop Minute
Start Datetime	Start Year Start Month Start Day Start Hour Start Minute
Stop Datetime	Stop Year Stop Month Stop Day Stop Hour Stop Minute
Weekday	Weekday bitmask

4.43.2.1.1 All parameters specified

If all Scheduling parameters are specified in a TimeRange, the Time Range MUST be active when the following condition is fulfilled:

- Daily Start Time <= Current time < Daily Stop Time
- Start Datetime <= Current datetime <= Stop Datetime
- Current weekday in (weekday bitmask)

For example, if Daily Start Time is set to 08:30, Daily Stop Time is set to 17:00, Start Datetime is set to 10th of January 2019 at 12:00, Stop Datetime is set to 5th of February 2019 at 12:00, Weekday bitmask is set to Monday to Friday, then the Time Range will be active:

- On the 10th of January 2019 from 12:00 to 17:00
- On subsequent weekdays until the 4th of February 2019, between 08:30 and 17:00
- On the 5th of February 2019 from 08:30 to 12:00

4.43.2.1.2 Some parameters undefined in parameter groups

If some of the parameters within a group are undefined, they are to be interpreted as wildcard and excluded from the comparison for the parameter group. For example, if Start Datetime has no Start Year and Start Month defined, but only Start Day set to 15, the time range MUST start on the 15th of every month also regardless of the year.

If Start Day is set to 15 and Stop Minute is set to 30, then the Time Range MUST be active every hour during 30 minutes from the 15th until the end of each month.

4.43.2.1.3 All parameters missing from a parameter group

If all fields are undefined (not in use) for a parameter group, the corresponding parameter group MUST NOT be used for the Time Range activation. For example, if none of the Stop Datetime parameters are defined, then the Time Range MUST be active when:

- Daily Start Time <= Current time < Daily Stop Time
- Start Datetime <= Current datetime
- Current weekday in (weekday bitmask)

1.1.4 Generic Schedule Time Range Get Command

This command is used to request the configuration of a Time Range ID for a supporting node.

The Generic Schedule Time Range Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_GENERIC_SCHEDULE							
Command = GENERIC_SCHEDULE_TIME_RANGE_GET							
Res	Time Range ID 1 (MSB)						
Time Range ID 2 (LSB)							

Time Range ID (16 bits)

This field is used to specify the requested Time Range ID.

The first byte MUST carry the most significant byte of the 15 bits.

A supporting node MUST return this value in the *Time Range ID* field in the returned Generic Schedule Report Command.

1.1.5 Generic Schedule Time Range Report Command

This command is used to advertise the configuration of a Time Range ID for a supporting node.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_GENERIC_SCHEDULE							
Command = GENERIC_SCHEDULE_TIME_RANGE_REPORT							
Res	Time Range ID 1 (MSB)						
Time Range ID 2 (LSB)							
In use	Weekday bitmask						
In use	Start Year (MSB)						
Start Year (LSB)							
In use	Stop Year (MSB)						
Stop Year (LSB)							
In use	Reserved			Start Month			
In use	Reserved			Stop Month			
In use	Reserved		Start Day				
In use	Reserved		Stop Day				
In use	Reserved		Start Hour				
In use	Reserved		Stop Hour				
In use	Res	Start Minute					
In use	Res	Stop Minute					
In use	Reserved		Daily Start Hour				
In use	Reserved		Daily Stop Hour				
In use	Res	Daily Start Minute					
In use	Res	Daily Stop Minute					
Res	Next Time Range ID						
Next Time Range ID							

Fields not described below MUST remain identical to the description in 1.1.3 Generic Schedule Time Range Set Command.

Res / Reserved

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Time Range ID (15 bits)

This field is used to indicate which Time Range ID configuration is being advertised.

If this field is set to a non-supported or non-configured Time Range ID, all scheduling parameters fields MUST be set to 0.

Next Time Range ID (15 bits)

This field is used to indicate the next non-empty Time Range ID.

This field MUST indicate the next non-empty Time Range ID. The value 0 MUST indicate that there is no following Time Range ID configured at the receiving node.

Non-empty Time Range slot ID means that at least 1 *In use* flag is set to 1.

1.1.6 Generic Schedule Set Command

This command is used to configure a Schedule for a supporting node.

A schedule MUST be active when:

- At least one of the included Time Ranges is active
- None of the excluded Time Ranges are active.

Refer to 4.43.2.2 for details.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_GENERIC_SCHEDULE							
Command = GENERIC_SCHEDULE_SET							
Schedule ID 1 (MSB)							
Schedule ID 2 (LSB)							
Number of Time Range IDs							
Include/ exclude	Time Range ID 1 (MSB)						
Time Range ID 1 (LSB)							
…							
Include/ exclude	Time Range ID N (MSB)						
Time Range ID N (LSB)							

Res / Reserved

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Schedule ID (16 bits)

This field is used to indicate which Schedule slot ID to use for the schedule being set.

This field MUST be in the range 0..[Number of supported Schedule IDs].

The value 0 MUST indicate that the supporting node MUST erase all configured schedules. In this case, the *Number of Time Range IDs* field MUST be set to 0 by a sending node.
Values in the range 1..[Number of supported Schedule IDs] MUST indicate the actual Schedule ID to configure.

If a non-supported Schedule ID is specified in this command, a receiving node MUST ignore the command.

Number of Time Range IDs (8 bits)

This field is used to indicate the number of Time Range IDs included as part of the Schedule.

This field MUST be in the range 0..[Number of supported Time Range IDs per Schedule].

A supporting node MUST ignore this command if it is set to a value higher than the reported *Number of supported Time Ranges per Schedule* field in the Generic Schedule Capabilities Report Command

The number of *Include/exclude* and *Time Range ID* field blocks in this command MUST be according to this field.

Include/exclude (N x 1 bit)

This field is used to indicate if the Schedule ID must include or exclude the corresponding Time Range ID.

The value 0 MUST indicate that the corresponding Time Range is excluded and the schedule MUST be active outside of the Time Range.
The value 1 MUST indicate that the corresponding Time Range is included and the schedule MUST be active during the defined Time Range.

Time Range ID (N x 15 bits)

This field is used to indicate which Time Range ID MUST be part of the Schedule ID.

Undefined Time Range IDs (all zeroed out) may be part of Schedules.

4.43.2.2 Schedules and Time Ranges examples

4.43.2.2.1 Example 1

- Time Range 1 is from 24th to 26th of December
- Time Range 2 is from 17:00 to 08:00 every day
- Time Range 3 is weekend days.

If a schedule is set to:

- Include Time Range 1
- Exclude Time Range 2
- Include Time Range 3

Then it MUST be active:

- every week-end from 08:00 to 17:00

- on the 24th,25th and 26th of December from 08:00 to 17:00 regardless of the current weekday.

4.43.2.2.2 Example 2

- Time Range 3 is undefined (all unused value).

If a schedule is set to:

- Include Time Range 1
- Include Time Range 2
- Include Time Range 3

Then the schedule MUST be active at all times.

4.43.2.2.3 Example 3

- Time Range 3 is undefined (all unused value).

If a schedule is set to:

- Include Time Range 1
- Exclude Time Range 2
- Exclude Time Range 3

Then the schedule MUST never be active.

4.43.2.2.4 Example 4

- Time Range 1 has no start time and finishes on the 30th of June
- Time Range 2 is from 08:00 to 17:00 on week days
- Time Range 3 is 10:00 to 15:00 on weekend days

If a schedule is set to:

- Include Time Range 1
- Include Time Range 2
- Include Time Range 3

Then the schedule MUST be active:

- All the time from the 1st of January until the 30th of June
- From 08:00 to 17:00 on weekdays between the 1st of July and the 31st of December every year
- From 10:00 to 15:00 on weekends between the 1st of July and the 31st of December every year

1.1.7 Generic Schedule Get Command

This command is used to request the configuration of a Schedule ID for a supporting node.

The Generic Schedule Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_GENERIC_SCHEDULE							
Command = GENERIC_SCHEDULE_GET							
Schedule ID 1 (MSB)							
Schedule ID 2 (LSB)							

Schedule ID (16 bits)

This field is used to specify the requested Schedule ID.

The first byte MUST carry the most significant byte of the 16 bits.

A supporting node MUST return this value in the *Schedule ID* field in the returned Generic Schedule Report Command.

1.1.8 Generic Schedule Report Command

This command is used to advertise the contents of a Schedule for a supporting node.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_GENERIC_SCHEDULE							
Command = GENERIC_SCHEDULE_REPORT							
Schedule ID 1 (MSB)							
Schedule ID 2 (LSB)							
Number of Time Range IDs							
Include/ exclude	Time Range ID 1 (MSB)						
Time Range ID 1 (LSB)							
…							
Include/ exclude	Time Range ID N (MSB)						
Time Range ID N (LSB)							
Next Schedule ID 1 (MSB)							
Next Schedule ID 2 (LSB)							

Fields not described below MUST remain identical to the description in 1.1.8 Generic Schedule Report Command.

Res / Reserved

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Schedule ID (16 bits)

This field is used to indicate which Schedule ID configuration is being advertised.

If this field is set to a non-supported or non-configured Schedule ID, all the *Number of Time Range IDs* field MUST be set to 0.

Number of Time Range IDs (8 bits)

This field is used to indicate the number of Time Range IDs included as part of the Schedule.

This field MUST be in the range 0..[Number of supported Time Range IDs per Schedule].

The number of *Include/exclude* and *Time Range ID* field blocks in this command MUST be according to this field.

Include/exclude (N x 1 bit)

This field is used to indicate if the Schedule ID must include or exclude the corresponding Time Range ID.

The value 0 MUST indicate that the corresponding Time Range is excluded and the schedule MUST be active outside of the Time Range.
The value 1 MUST indicate that the corresponding Time Range is included and the schedule MUST be active during the defined Time Range.

Time Range ID (N x 15 bits)

This field is used to indicate which Time Range ID MUST be part of the Schedule ID.

Undefined Time Range IDs (all zeroed out) may be part of Schedules.

Next Schedule ID (16 bits)

This field is used to indicate the next non-empty Schedule ID.

This field MUST indicate the next non-empty Schedule slot ID. The value 0 MUST indicate that there is no following Schedule Slot ID configured at the supporting node.

4.44 Geographic Location Command Class, version 1

The Geographic Location Command Class is used to read latitude and longitude from another device. The latitude and longitude may also be set according to the geographic location in question. Date and geographic location may be used to calculate sunrise and sunset for e.g. automatic lighting control.

4.44.1 Multi Channel Considerations

Multi Channel End Points SHOULD NOT support the Geographic Location Command Class.

4.44.2 Geographic Location Set Command

This command is 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.

4.44.3 Geographic Location Get Command

This command is used to request latitude and longitude from a device.

The Geographic Location Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

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

4.44.4 Geographic Location Report Command

This command returns 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_REPORT							
Longitude Degrees							
Long. Sign	Longitude Minutes						
Latitude Degrees							
Lat. Sign	Latitude Minutes						

Refer to description under the Geographic Location Set Command.

4.45 HRV Status Command Class, version 1

The residential Heat Recovery Ventilation (HRV) Status Command Class is used to read out a number of parameters in the ventilation system.

4.45.1 HRV Status Get Command

This command is used to request specific parameters from the ventilation system.

The HRV Status Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

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.

Value	Status Parameter
0x00	Outdoor Air temperature
0x01	Supply Air (to room) temperature
0x02	Exhaust Air (from room) temperature
0x03	Discharge Air temperature
0x04	Room temperature
0x05	Relative Humidity in room
0x06	Remaining filter life

4.45.2 HRV Status Report Command

This command is 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							

Status Parameter (8 bits)

The status parameter used to indicate which status parameter that is reported. Refer to 4.45.1 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

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 (N bytes)

The value MUST be treated as a signed field (Table 10). The value MAY be 1, 2 or 4 bytes in size. This first byte is the most significant byte.

4.45.3 HRV Status Supported Get Command

This command is used to request a bitmap of the supported status parameters.

The HRV Status Supported Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_HRV_STATUS							
Command = HRV_STATUS_SUPPORTED_GET							

4.45.4 HRV Status Supported Report Command

This command is 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							

Bit Mask (N bytes)

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 4.45.1 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 MUST be determined from the length field in the frame.

4.46 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.

4.46.1 HRV Mode Set

This command is 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 MAY 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 may be used to manually control the device.

All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

Reserved

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

4.46.2 HRV Mode Get Command

This command is used to request the current mode from the device.

The HRV Mode Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_HRV_CONTROL							
Command = HRV_CONTROL_MODE_GET							

4.46.3 HRV Mode Report Command

The HRV Mode Report Command is used to report the mode from the device.

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

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

4.46.4 HRV Bypass Set Command

This command is 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 MAY be 0x00 (close) or 0xFF (open).

Furthermore, the field MAY carry 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 values SHOULD 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) MAY be used to set the bypass into automatic mode.

4.46.5 HRV Bypass Get Command

This command is used to request the current bypass setting.

The HRV Bypass Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_HRV_CONTROL							
Command = HRV_CONTROL_BYPASS _GET							

4.46.6 HRV Bypass Report Command

This command is 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 in 4.46.4 HRV Bypass Set.

4.46.7 HRV Ventilation Rate Set Command

This command is 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 MAY be 0x00 (off) or 0xFF (on).

The field MAY carry a percentage value between 1 to 99 (0x01 - 0x63). A ventilation system MAY map the values 1..99 to less than 99 steps.

4.46.8 HRV Ventilation Rate Get Command

This command is used to request the current ventilation rate setting.

The HRV Ventilation Rate Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_HRV_CONTROL							
Command = HRV_CONTROL_VENTILATION_RATE_GET							

4.46.9 HRV Ventilation Rate Report Command

This command is 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 4.46.7 HRV Ventilation Rate Set Command.

4.46.10 HRV Mode Supported Get Command

This command is used to request the supported modes from the device.

The HRV Mode Supported Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_HRV_CONTROL							
Command = HRV_CONTROL_MODE_SUPPORTED_GET							

4.46.11 HRV Mode Supported Report Command

This command is used to report 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_REPORT							
Reserved				Manual Control Supported			
Bit Mask 1							
…							
Bit Mask N							

Reserved

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

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, 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 (N bytes)

The Bit Mask fields describe the supported modes by the device.

- Bit 0 in Bit Mask 1 field indicates if Mode = 0 (Off) is supported.
- Bit 1 in Bit Mask 1 field indicates if Mode = 1 (Demand / Automatic) is supported.
- …

If the Mode is supported the bit MUST be set to 1. If the Mode is not supported the bit MUST be set to 0. 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 MUST be determined from the length field in the frame.

All available modes are given in Section 4.46.1 HRV Mode Set.

4.47 Humidity Control Mode Command Class, version 1-2

The Humidity Control Mode Command Class is used to control a humidity control device.

4.47.1 Humidity Control Mode Set Command

This command is used to set the humidity control mode in the device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_HUMIDITY_CONTROL_MODE							
Command = HUMIDITY_CONTROL_MODE_SET							
Reserved				Mode			

Reserved

The reserved field is for future use. Reserved bits MUST be set to zero by a sending node. Reserved bits MUST be ignored by a receiving node.

Mode (4 bits)

The encoding of the humidity control mode field MUST be according to Table 56.

Table 56, Mode

Mode	Description	Version
0	Off – Humidity control system is off.	1
1	Humidify – The system will attempt to raise humidity to the humidifier setpoint.	1
2	De-humidify – The system will attempt to lower the humidity to the de-humidifier setpoint.	1
3	Auto - The system will automatically switch between humidifying and de-humidifying in order to satisfy the humidify and de-humidify setpoints	2

All other values are reserved. Reserved values MUST NOT be used by a sending node and MUST be ignored by a receiving node.

4.47.2 Humidity Control Mode Get Command

This command is used to request the supported humidity control modes from the device.

The Humidity Control Mode Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_HUMIDITY_CONTROL_MODE							
Command = HUMIDITY_CONTROL_MODE_GET							

4.47.3 Humidity Control Mode Report Command

This command is used to report the humidity control mode from the device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_HUMIDITY_CONTROL_MODE							
Command = HUMIDITY_CONTROL_MODE_REPORT							
Reserved				Mode			

Reserved

The reserved field is for future use. Reserved bits MUST be set to zero by a sending node. Reserved bits MUST be ignored by a receiving node.

Mode (4 bits)

The encoding of the humidity control mode field MUST be according to Table 56.

4.47.4 Humidity Control Mode Supported Get Command

This command is used to request the supported modes from the device.

The Humidity Control Mode Supported Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_HUMIDITY_CONTROL_MODE							
Command = HUMIDITY_CONTROL_MODE_SUPPORTED_GET							

4.47.5 Humidity Control Mode Supported Report Command

This command is used to report the supported humidity control modes from the device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_HUMIDITY_CONTROL_MODE							
Command = HUMIDITY_CONTROL_MODE_SUPPORTED_REPORT							
Bit Mask 1							
…							
Bit Mask N							

Bit Mask (N bytes)

The Bit Mask fields MUST advertise the supported humidity control modes by the device. The encoding of the Size field MUST be according to Table 57.

Table 57, Supported Humidity Control Mode

Field	Bit	Support for Control Mode	Version
Bit Mask 1	0	*Reserved*	1
Bit Mask 1	1	1 (Humidifier)	1
Bit Mask 1	2	2 (De-humidifer)	1
Bit Mask 1	3	3 (Auto)	2

All Bit Mask fields and bits not specified above are reserved. Reserved bits MUST be set to zero by a sending node. Reserved bits MUST be ignored by a receiving node.

For each individual bit, The value ‘0’ MUST signify that the actual Mode is not supported.
The value ‘1’ MUST signify that the actual Mode is supported.

A sending node MAY omit trailing Bit Mask fields if they are not needed. The number of Bit Mask fields MUST be determined from the length field in the frame.

4.48 Humidity Control Operating State Command Class, version 1

The Humidity Control Operating State Command Class is used to obtain the operating state of the humidity control device.

4.48.1 Humidity Control Operating State Get Command

This command is used to request the operating state of the humidity control device.

The Humidity Control Operating State Report Command MUST be returned in response to thiscommand.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_HUMIDITY_CONTROL_OPERATING_STATE							
Command = HUMIDITY_CONTROL_OPERATING_STATE_GET							

4.48.2 Humidity Control Operating State Report Command

This command is used to report the operating state of the humidity control device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_HUMIDITY_CONTROL_OPERATING_STATE							
Command = HUMIDITY_CONTROL_OPERATING_STATE_REPORT							
Reserved				Operating State			

Reserved

The reserved field is for future use. Reserved bits MUST be set to zero by a sending node. Reserved bits MUST be ignored by a receiving node.

Operating State (4 bits)

The Operating State field MUST advertise the current operating state.

The encoding of the Operating State field MUST be according to Table 58.

Table 58, Operating State

Operating State	Description	Version
0	Idle	1
1	Humidifying	1
2	De-humidifying	1

All other values are reserved. Reserved values MUST NOT be used by a sending node and MUST be ignored by a receiving node.

4.49 Humidity Control Setpoint Command Class, version 1-2

The Humidity Control Setpoint Command Class is used for humidity control setpoint handling.

4.49.1 Humidity Control Setpoint Set Command

This command is used to set the humidity control setpoint in the device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_HUMIDITY_CONTROL_SETPOINT							
Command = HUMIDITY_CONTROL_SETPOINT_SET							
Reserved				Setpoint Type			
Precision			Scale		Size		
Value 1							
…							
Value N							

Reserved

The reserved field is for future use. Reserved bits MUST be set to zero by a sending node. Reserved bits MUST be ignored by a receiving node.

Setpoint Type (4 bits)

The Setpoint Type field MUST specify the humidity control setpoint to be set in the humidity control device. The encoding of the Setpoint Type field MUST be according to Table 59.

Table 59, Setpoint Type

Setpoint Type	Description	Version
1	Humidifier	1
2	De-humidifier	1
3	Auto	2

All other values are reserved. Reserved values MUST NOT be used by a sending node and MUST be ignored by a receiving node.

Precision (3 bits)

The precision field MUST specify the precision of the setpoint value. The number indicates the number of decimals. As an example, the decimal value 1025 with precision two (2) is equal to 10.25.

Scale (2 bits)

The Scale field MUST specify the humidity scale used. The encoding of the Scale field MUST be according to Table 60.

Table 60, Scale values

Scale	Description	Version
0	Percentage value	1
1	Absolute humidity (g/m^3)	1

All other values are reserved. Reserved values MUST NOT be used by a sending node and MUST be ignored by a receiving node.

Size (3 bits)

The Size field MUST specify the number of bytes used for the Value field. The encoding of the Size field MUST be according to Table 61.

Table 61, Size values

Size	Description	Version
1	Value field is 1 byte long	1
2	Value field is 2 bytes long	1
4	Value field is 4 bytes long	1

All other values are reserved. Reserved values MUST NOT be used by a sending node and MUST be ignored by a receiving node.

Value (N bytes)

The value is a signed field. The field MAY be one, two, or four bytes in size as specified by the Size field. Value 1 is the most significant byte.

4.49.2 Humidity Control Setpoint Get Command

This command is used to request the given humidity control setpoint type in a device.

The Humidity Control Setpoint Report MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_HUMIDITY_CONTROL_SETPOINT							
Command = HUMIDITY_CONTROL_SETPOINT_GET							
Reserved				Setpoint Type			

Reserved

The reserved field is for future use. Reserved bits MUST be set to zero by a sending node. Reserved bits MUST be ignored by a receiving node.

Setpoint Type (4 bits)

This field MUST indicate which humidity control setpoint MUST be returned by the supporting node. The encoding of the Setpoint Type field MUST be according to Table 59.

4.49.3 Humidity Control Setpoint Report Command

This command is used to report the value of the humidity control setpoint type in the device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_HUMIDITY_CONTROL_SETPOINT							
Command = HUMIDITY_CONTROL_SETPOINT_REPORT							
Reserved				Setpoint Type			
Precision			Scale		Size		
Value 1							
…							
Value N							

Reserved

The reserved field is for future use. Reserved bits MUST be set to zero by a sending node. Reserved bits MUST be ignored by a receiving node.

For fields' description, refer to 4.49.1 Humidity Control Setpoint Set Command

4.49.4 Humidity Control Setpoint Supported Get Command

This command is used to request the humidity control setpoint types supported by the device.

The Humidity Control Setpoint Supported Report Command MUST be returned in response to a this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_HUMIDITY_CONTROL_SETPOINT							
Command = HUMIDITY_CONTROL_SETPOINT_SUPPORTED_GET							

4.49.5 Humidity Control Setpoint Supported Report Command

This command is used to report the humidity control setpoint types supported by the device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_HUMIDITY_CONTROL_SETPOINT							
Command = HUMIDITY_CONTROL_SETPOINT_SUPPORTED_REPORT							
Bit Mask 1							
…							
Bit Mask N							

Bit Mask (N bytes)

The Bit Mask fields MUST advertise the humidity control setpoint types supported by the device.The encoding of the Size field MUST be according to Table 62.

Table 62, Humidity Control Setpoint Supported Report values

Field	Bit	Support for Setpoint Type	Version
Bit Mask 1	0	*Reserved*	1
Bit Mask 1	1	1 (Humidifier)	1
Bit Mask 1	2	2 (De-humidifer)	1
Bit Mask 1	3	3 (Auto)	2

All Bit Mask fields and bits not specified above are reserved. Reserved bits MUST be set to zero by a sending node. Reserved bits MUST be ignored by a receiving node.

For each individual bit, The value ‘0’ MUST signify that the actual Setpoint Type is not supported. The value ‘1’ MUST signify that the actual Setpoint Type is supported.

A sending node MAY omit trailing Bit Mask fields if they are not needed. The number of Bit Mask fields MUST be determined from the length field in the frame.

4.49.6 Humidity Control Setpoint Scale Supported Get Command

This command is used to retrieve the supported scales of the humidity control setpoints in the device.

The Humidity Control Scale Supported Report Command MUST be returned in response to this command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_HUMIDITY_CONTROL_SETPOINT							
Command = HUMIDITY_CONTROL_SETPOINT_SCALE_SUPPORTED_GET							
Reserved				Setpoint Type			

Reserved

The reserved field is for future use. Reserved bits MUST be set to zero by a sending node. Reserved bits MUST be ignored by a receiving node.

Setpoint Type (4 bits)

Refer to the description under the Humidity Control Setpoint Set Command.

4.49.7 Humidity Control Setpoint Scale Supported Report Command

This command indicates the supported scales of the humidity control setpoint in a bit mask format.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_HUMIDITY_CONTROL_SETPOINT							
Command = HUMIDITY_CONTROL_SETPOINT_SCALE_SUPPORTED_REPORT							
Reserved				Scale Bit Mask			

Reserved

The reserved field is for future use. Reserved bits MUST be set to zero by a sending node. Reserved bits MUST be ignored by a receiving node.

Scale Bit Mask (4 bits)

Refer to the description under the Humidity Control Setpoint Set Command.

4.49.8 Humidity Control Setpoint Capabilities Get Command

This command is used to request the minimum and maximum setpoint values for a given humidity control Setpoint Type.

The Humidity Control Setpoint Capabilities Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_HUMIDITY_CONTROL_SETPOINT							
Command = HUMIDITY_CONTROL_SETPOINT_CAPABILITIES_GET							
Reserved				Setpoint Type			

Reserved

The reserved field is for future use. Reserved bits MUST be set to zero by a sending node. Reserved bits MUST be ignored by a receiving node.

Setpoint Type (4 bits)

Refer to the description under the Humidity Control Setpoint Set Command.

4.49.9 Humidity Control Setpoint Capabilities Report Command

This command is used to report the minimum and maximum values of the requested humidity control setpoint type.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_HUMIDITY_CONTROL_SETPOINT							
Command = HUMIDITY_CONTROL_SETPOINT_CAPABILITIES_REPORT							
Reserved				Setpoint Type			
Precision			Scale		Size		
Minimum Value 1							
…							
Minimum Value N							
Precision			Scale		Size		
Maximum Value 1							
…							
Maximum Value N							

Refer to the description under the Humidity Control Setpoint Set Command for other parameter/field descriptions.

4.50 IR Repeater Command Class, version 1

The IR Repeater Command Class is used to get supporting nodes to learn and/or repeat IR signals used to control appliances.

A supporting node MUST have the capability to issue or receive IR signals.

1.1.1 IR Repeater Capabilities Get Command

This command is used to request the IR Repeater capabilities of a node.

The IR Repeater Capabilities Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_IR_REPEATER (0xA0)							
Command = IR_REPEATER_CAPABILITIES_GET (0x01)							

1.1.2 IR Repeater Capabilities Report Command

This command is used to advertise the IR Repeater capabilities of the sending node.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_IR_REPEATER (0xA0)							
Command = IR_REPEATER_CAPABILITIES_REPORT (0x02)							
Number of IR Code identifiers for learning							
IRR	LCR	RLC	PES	BES	Duty Cycle bitmask		
Carrier							
Minimum Sub Carrier							
Maxium Sub Carrier							
Minimum Pulse time unit (MSB)							
Minimum Pulse time unit (LSB)				Maximum Pulse time unit (MSB)			
Maximum Pulse time unit (LSB)							

Reserved / Res

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Number of IR Code identifiers for learning (8 bits)

This field indicates how many slots the sending node supports for learning and repeating IR Codes.

The value 0 MUST indicate that the no code can be learned and repeated.
Values in the range 1..255 MUST indicate the number of codes that can be learnt and repeated.

A node advertising a value greater than 0 MUST support the following commands:

- IR Repeater IR Code Learning Start Command
- IR Repeater IR Code Learning Stop Command
- IR Repeater IR Code Learning Status Command
- IR Repeater Learnt IR Code Remove Command
- IR Repeater Learnt IR Code Get Command
- IR Repeater Learnt IR Code Report Command

A node advertising a value greater than 0 MUST set the *LCR* field and/or *RLC* field to 1.

IRR (IR Repeater Repeat Command) (1 bit)

This field indicates if the nodes supports the following commands:

- IR Repeater Repeat Command

The value 0 MUST indicate that the commands above are not supported.
The value 1 MUST indicate that the commands above are supported.

LCR (Learnt Code Readback) (1 bit)

This field indicates if the supporting node can be read back learnt codes, so that a controlling node can read back current learnt codes. This field indicates if the nodes supports the following commands:

- IR Repeater Learnt IR Code Readback Get Command
- IR Repeater Learnt IR Code Readback Report Command

The value 0 MUST indicate that the commands above are not supported.
The value 1 MUST indicate that the commands above are supported.

If this field is set to 1, the *Number of IR Code Identifiers for learning* field MUST be set to a value greater than 0.

If this field is set to 1, the *IRR* field MUST also be set to 1.

RLC (Repeat Learnt Code) (1 bit)

This field indicates if the supporting node can be repeat learnt codes. This field indicates if the nodes supports the following commands:

- IR Repeater Repeat Learnt Code Command

The value 0 MUST indicate that the command above is not supported.
The value 1 MUST indicate that the command above is supported.

If this field is set to 1, the *Number of IR Code Identifiers for learning* field MUST be set to a value greater than 0.

If this field is set to 1, the *IRR* field MUST also be set to 1.

PES (Pulse Encoding Support) (1 bit)

This field indicates if the supporting node supports Pulse Encoding of the repeat and/or learn codes.

The value 0 MUST indicate that Pulse Encoding of IR signals is not supported.
The value 1 MUST indicate that Pulse Encoding of IR signals is supported.

If this field is set to 0, the *BES* field MUST be set to 1.

BES (Bit Encoding Support) (1 bit)

This field indicates if the supporting node supports Bit Encoding of the repeat and/or learn codes.

The value 0 MUST indicate that Bit Encoding of IR signals is not supported.
The value 1 MUST indicate that Bit Encoding of IR signals is supported.

If this field is set to 0, the *PES* field MUST be set to 1.

If this field is set to 1, the sending node MUST support the following commands:

- IR Repeater Configuration Set Command
- IR Repeater Configuration Get Command
- IR Repeater Configuration Report Command.

Duty Cycle bitmask (3 bits)

This field is used to advertise the supported Duty Cycles.

This field MUST be treated as a bitmask and encoded as follows:

- Bit 0 in Duty Cycle field MUST represent ‘1/2 Duty Cycle’.
- Bit 1 in Duty Cycle field MUST represent ‘1/3 Duty Cycle’.
- Bit 2 in Duty Cycle field MUST represents ‘1/4 Duty Cycle’.

If a Duty Cycle is supported, the corresponding bit MUST be set to ‘1’.
If a Duty Cycle is not supported, the corresponding bit MUST be set to ‘0’.

Carrier (8 bits)

This field is used to indicate the IR Carrier that the supporting node will use for repeating IR Codes indicated when receiving the IR Repeater Repeat Command.

This field MUST be expressed as 820 + x nm (nanometers). The values 0x00..0xFF MUST represents 820nm..1075nm.

Minimum Sub Carrier (8 bits)

This field MUST indicates the lower limit value that can be set at the supporting node in the IR Repeater Repeat Command for the *Sub Carrier* field.

This field MUST be expressed in kHz (kilo Hertz).

Maximum Sub Carrier (8 bits)

This field MUST indicates the upper limit value that can be set at the supporting node in the IR Repeater Repeat Command for the *Sub Carrie*r field.

This field MUST be expressed in kHz (kilo Hertz).

Minimum Pulse time Unit (12 bits)

This field MUST indicates the lower limit value that can be set at the supporting node in the IR Repeater IR Code Learning Start Command and IR Repeater Repeat Command for the *Pulse time unit* field.

If the *PES* field is set to 0, this field MUST set to 0.

Maximum Pulse time Unit (12 bits)

This field MUST indicates the upper limit value that can be set at the supporting node in the IR Repeater IR Code Learning Start Command and IR Repeater Repeat Command for the *Pulse time unit* field.

If the *PES* field is set to 0, this field MUST set to 0.

1.1.3 IR Repeater IR Code Learning Start Command

This command is used to instruct a supporting node to start learning an IR Code.

This command MUST be ignored by a receiving node if it advertises 0 *Number of IR Code identifiers for learning* in the IR Repeater Capabilities Report Command.

A supporting node MUST stop learning IR Code and return a IR Repeater IR Code Learning Status Command a when a code has been received/learnt.

This command MUST be ignored if the supporting node is already learning an IR code.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_IR_REPEATER (0xA0)							
Command = IR_REPEATER_IR_CODE_LEARNING_START (0x03)							
IR Code identifier							
Timeout							
Reserved				Pulse time unit (MSB)			
Pulse time unit (LSB)							

IR Code identifier (8 bits)

This field indicates which IR Code identifier to use for storing the IR Code that will be learnt.
Values in the range 1..255 MUST indicate the actual IR Code identifier to use for learning a new code.

This command MUST be ignored if this field is set to 0 or a larger value than the *Number of IR Code identifiers for learning* field advertised in the IR Repeater Capabilities Report Command.

Timeout (8 bits)

This field is used to indicate how long the supporting node MUST wait for an IR code to learn.

If no IR code is learnt within the time in seconds indicated by this field, a supporting node MUST stop learning codes and return a IR Repeater IR Code Learning Status Command with the *status* field set to 0x02 (Failed:timeout).

Pulse time unit (12 bits)

This field is used to configure the IR Code learning Pulse Encoding time unit.

This field MUST be expressed in (µs) and be in the range 1..4096.

Values lower than the *Minimum Pulse time unit* advertised by the supporting node in the IR Repeater Capabilities Report Command MUST be ignored by a supporting node.

Values higher than the *Maximum Pulse time unit* advertised by the supporting node in the IR Repeater Capabilities Report Command MUST be ignored by a supporting node.

1.1.4 IR Repeater IR Code Learning Stop Command

This command is used to instruct a supporting node to stop learning an IR Code.

This command MUST be ignored by a receiving node if it advertises 0 *Number of IR Code identifiers for learning* in the IR Repeater Capabilities Report Command.

A supporting node MUST stop learning IR Code if a learning process was ongoing when this command is received.

A supporting node MUST return a IR Repeater IR Code Learning Status Command in response to this command.

7	6	5	4	3	2	1		0
Command Class = COMMAND_CLASS_IR_REPEATER (0xA0)								
Command = IR_REPEATER_IR_CODE_LEARNING_STOP (0x04)								

1.1.5 IR Repeater IR Code Learning Status Command

This command is used to report the status of the last IR Code learning operation. It MUST be issued by a supporting node when it stops learning IR Codes.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_IR_REPEATER (0xA0)							
Command = IR_REPEATER_IR_CODE_LEARNING_STATUS (0x05)							
IR Code identifier							
Status							

IR Code identifier (8 bits)

This field indicates which IR Code identifier has been used for storing the IR Code during IR Code learning.

If a learning process was completed, interrupted or timed out, this field MUST be in the range 1..255 and MUST be set to the same value as the *IR Code identifier* field of the IR Repeater IR Code Learning Start Command that initiated IR Code learning.

If a no learning process was ongoing and a IR Repeater IR Code Learning Stop Command was received, this field MUST be set to 0.

Status(8 bit)

This field is used to indicate the status of the IR Code learning operation. This field MUST encoded according to Table 63.

Table 63,IR Code Learning Status::Status encoding

Value	Description
0x00	Failed: Learning buffer overflow The node learning buffer is too small to learnt the entire IR Code.
0x01	Failed: Forced learning stop The node has received an IR Repeater IR Code Learning Stop Command before any code was learnt
0x02	Failed: timeout The node has timed out and did not learn any code
0xFF	Success A new code was learnt.

1.1.6 IR Repeater Learnt IR Code Remove Command

This command is used to instruct a supporting node to erase a previously learnt an IR Code.

This command MUST be ignored by a receiving node if it advertises 0 *Number of IR Code identifiers for learning* in the IR Repeater Capabilities Report Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_IR_REPEATER (0xA0)							
Command = IR_REPEATER_LEARNT_IR_CODE_REMOVE (0x06)							
IR Code identifier							

IR Code identifier (2 bytes)

This field indicates which IR Code identifier to erase.
Values in the range 1..255 MUST indicate the actual IR Code identifier that MUST be erased by the supporting node.

The value 0 MUST indicate to erase all IR codes.

This command MUST be ignored if this field is set to a larger value than the *Number of IR Code identifiers for learning* field advertised in the IR Repeater Capabilities Report Command.

1.1.7 IR Repeater Learnt IR Code Get Command

This command is used to request a supporting node which IR Code identifiers have a valid IR Code learnt.

This command MUST be ignored by a receiving node if it advertises 0 *Number of IR Code identifiers for learning* in the IR Repeater Capabilities Report Command.

The IR Repeater Learnt IR Code Report Command MUST be returned in response to this command if it was not ignored.

This command MUST NOT be issued via multicast addressing.

A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_IR_REPEATER (0xA0)							
Command = IR_REPEATER_LEARNT_IR_CODE_GET (0x07)							

1.1.8 IR Repeater Learnt IR Code Report Command

This command is used to advertise the current IR learnt code stored at the supporting node.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_IR_REPEATER (0xA0)							
Command = IR_REPEATER_LEARNT_IR_CODE_REPORT (0x08)							
Reserved			IR Code idenfier status length				
IR Code identifier status bitmask 1							
...							
IR Code identifier status bitmask N							

IR Code idenfier status length (5 bits)

This field is used to indicate the length in bytes of the *IR Code identifier status bitmask* field.

This field MUST be set to the minimum value which allows advertising all IR Code currently set at the supporting node.

IR Code identifier status bitmask (N bytes)

This field advertises the supported IR Code identifier status values. The length of this field in bytes MUST match the value advertised in the *IR Code Identifier Status Length* field.

This field MUST be encoded as a bitmask:

- Bit 0 in Bit Mask 1 represents IR Code Identifier 0x01.
- Bit 1 in Bit Mask 1 represents IR Code Identifier 0x02.
- …

If an IR Code is stored for the IR Code Identifier, the corresponding bit MUST be set to ‘1’.
If no IR Code is stored for the IR Code Identifier, the corresponding bit MUST be set to ‘0’.

Only IR Codes with their bit set to ‘1’ can be repeated with the IR Repeater Repeat Learnt Code Command

1.1.9 IR Repeater Learnt IR Code Readback Get Command

This command is used to request a supporting node has learnt an IR Code to return the code details to a controlling node.

This command MUST be ignored by a receiving node if it advertises no support for *LCR (Learnt Code Readback)* in the IR Repeater Capabilities Report Command.

The IR Repeater Learnt IR Code Report Command MUST be returned in response to this command if it was not ignored.

This command MUST NOT be issued via multicast addressing.

A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_IR_REPEATER (0xA0)							
Command = IR_REPEATER_LEARNT_IR_CODE_READBACK_GET (0x09)							
IR Code identifier							
Res	Report Number (MSB)						
Report Number (LSB)							

IR Code identifier (8 bits)

This field is used to indicate the IR Code identifier that the supporting node MUST return in response to this command.

A supporting node receiving a not supported IR Code Identifier or current unset/empty IR Code Identifier MUST return a report with *Report Number* set to 0, *Last* set to 1 and no *Data block* field.

Report Number (15 bits)

This field indicates the data block sequence number of the requested Learnt IR Code. The first data block MUST be identified by the Report Number value 1.

If this field is set to 0 or a value higher than the total number of reports (indicated with the *last* field in the report), a supporting node MUST return a report with no *Data block* field.

1.1.10 IR Repeater Learnt IR Code Readback Report Command

This command is used to advertise the “contents” of a Learn IR Code.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_IR_REPEATER (0xA0)							
Command = IR_REPEATER_LEARNT_IR_CODE_READBACK_REPORT (0x0A)							
IR Code identifier							
Sub Carrier							
Reserved		Duty Cycle		Pulse time unit (MSB)			
Pulse time unit (LSB)							
Last	Report Number (MSB)						
Report Number (LSB)							
Data block type 1	Data block length 1						
Data block value 1, 1							
...							
Data block value 1, N							
...							
Data block type M	Data block length M						
Data block value M, 1							
...							
Data block value M, N							

IR Code identifier (8 bits)

This field is used to indicate the IR Code identifier that the supporting node is currently sending.

Sub Carrier (8 bits)

For field description, refer to 1.1.15 IR Repeater Repeat Command

Duty Cycle (2 bits)

For field description, refer to 1.1.15 IR Repeater Repeat Command

Pulse time unit (12 bits)

For field description, refer to 1.1.15 IR Repeater Repeat Command

Report Number(15 bits)

For field description, refer to 1.1.15 IR Repeater Repeat Command

Last (1 bit)

For field description, refer to 1.1.15 IR Repeater Repeat Command

Data block type (2 bits)

For field description, refer to 1.1.15 IR Repeater Repeat Command

Data Block Length (5 bits)

For field description, refer to 1.1.15 IR Repeater Repeat Command

Data Block value (N bytes)

For field description, refer to 1.1.15 IR Repeater Repeat Command

1.1.11 IR Repeater Configuration Set Command

This command is used to configure the IR signal properties before the supporting node would repeat IR Codes.

This command MUST be ignored by a receiving node if it advertises no support for *Bit Encoding Support (BES)* in the IR Repeater Capabilities Report Command.

A supporting node MUST have a default bit 1, bit 0 and period configured after reset.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_IR_REPEATER (0xA0)							
Command = IR_REPEATER_CONFIGURATION_SET (0x0B)							
Period (MSB)							
Reserved				Period (LSB)			
Data bit 0 encoding length (MSB)							
Data bit 0 encoding length (LSB)							
Data bit 0 encoding 1							
...							
Data bit 0 encoding N							
Data bit 1 encoding length (MSB)							
Data bit 1 encoding length (LSB)							
Data bit 1 encoding 1							
...							
Data bit 1 encoding M							

Period (12 bits)

This field is used to indicate the time period used for data bit encoding.

This field MUST be expressed in micro seconds (µs). This field MUST be in the range 1..4096

Data bit ’0’ encoding length (2 bytes)

This field is used to indicate the length in bits of the *Data bit ’0’ encoding* field.

This field MUST be set to the exact number of bits to be interpreted from the *Data bit ’0’ encoding* field.

This field MUST be ignored by a supporting node if the *BES* field is set to 0 in the IR Repeater Capabilities Report Command.

Data bit ’0’ encoding (N bytes)

This field advertisesthe Hi/Low polarization encoding to apply to encode a 0 in an IR Code using the IR Repeater Repeat Command and IR Repeater Learnt IR Code Readback Report Command.

This field MUST be ignored by a supporting node if the *BES* field is set to 0 in the IR Repeater Capabilities Report Command.

The length of this field MUST be the minimum amount of full bytes allowing to have the number of bits indicated in the the *Data bit ’0’ encoding Length* field. .e.g if the *Data bit ’0’ encoding* field is set to 13, this field MUST be 2 bytes long.

Bits MUST be encoded as follows:

- The value 0 MUST indicate a Low polarization
- The value 1 MUST indicate a High polarization.

The whole sequence indicated in this field MUST be repeated by a supporting node when receiving a IR Repeater Repeat Command indicating to repeat a bit 0.

Data bit ’1’ encoding length (2 bytes)

This field is used to indicate the length in bits of the *Data bit ’1’ encoding* field.

This field MUST be set to the exact number of bits to be interpreted from the *Data bit ’1’ encoding* field.

This field MUST be ignored by a supporting node if the *BES* field is set to 0 in the IR Repeater Capabilities Report Command.

Data bit ’1’ encoding (N bytes)

This field advertisesthe Hi/Low polarization encoding to apply to encode a 1 in an IR Code using the IR Repeater Repeat Command and IR Repeater Learnt IR Code Readback Report Command.

This field MUST be ignored by a supporting node if the *BES* field is set to 0 in the IR Repeater Capabilities Report Command.

The length of this field MUST be the minimum amount of full bytes allowing to have the number of bits indicated in the the *Data bit ’1’ encoding Length* field. .e.g if the *Data bit ’1’ encoding* field is set to 13, this field MUST be 2 bytes long.

Bits MUST be encoded as follows:

- The value 0 MUST indicate a Low polarization
- The value 1 MUST indicate a High polarization.

The whole sequence indicated in this field MUST be repeated by a supporting node when receiving a IR Repeater Repeat Command indicating to repeat a bit 1.

1.1.12 IR Repeater Configuration Get Command

This command is used to request the IR signal configuration at the supporting node.

The IR Repeater Configuration Report Command MUST be returned in response to this command.

This command MUST be ignored by a receiving node if it advertises no support for *Bit Encoding Support (BES)* in the IR Repeater Capabilities Report Command.

This command MUST NOT be issued via multicast addressing.

A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_IR_REPEATER							
Command = IR_REPEATER_CONFIGURATION_GET (0x0C)							

1.1.13 IR Repeater Configuration Report Command

This command is used to advertise the current IR signal repeating configuration at the supporting node.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_IR_REPEATER (0xA0)							
Command = IR_REPEATER_CONFIGURATION_REPORT (0x0D)							
Period (MSB)							
Reserved				Period (LSB)			
Data bit 0 encoding length (MSB)							
Data bit 0 encoding length (LSB)							
Data bit 0 encoding 1							
...							
Data bit 0 encoding N							
Data bit 1 encoding length (MSB)							
Data bit 1 encoding length (LSB)							
Data bit 1 encoding 1							
...							
Data bit 1 encoding M							

For fields' description, refer to 1.1.11 IR Repeater Configuration Set Command

1.1.14 IR Repeater Repeat Learnt Code Command

This command is used to request the receiving node to send an IR Code.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_IR_REPEATER (0xA0)							
Command = IR_REPEATER_REPEAT_LEARNT_CODE (0x0E)							
IR Code identifier							

IR Code identifier (8 bits)

This field indicates which learnt IR Code identifier to repeat.
Values in the range 1..255 MUST indicate the actual IR Code identifier to use for repeating the learnt code.

This command MUST be ignored if this field is set to 0, if no IR Code is learnt for the received Identifier or if this field is set to a larger value than the *Number of IR Code identifiers for learning* field advertised in the IR Repeater Capabilities Report Command.

A supporting node MUST repeat the IR code if a code has been learnt for the actual IR Code identifier.

1.1.15 IR Repeater Repeat Command

This command is used to request the receiving node to send/repeat an IR Code.

This command MUST be ignored if the supporting node advertises no support for IR Code Repeating (*IRR*) in the IR Repeater Capabilities Report Command.

This command MUST NOT be issued via multicast addressing.

A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_IR_REPEATER (0xA0)							
Command = IR_REPEATER_REPEAT (0x0F)							
Sequence number							
Sub carrier							
Reserved		Duty Cycle		Pulse time unit (MSB)			
Pulse time unit (LSB)							
Last	Report number (MSB)						
Report number (LSB)							
Data block type 1		Data block length 1					
Data block value 1, 1							
...							
Data block value 1, N							
...							
Data block type M		Data block length M					
Data block value M, 1							
...							
Data block value M, N							

Sequence number (8 bits)

This field is used to keep track of the command that initiated the IR Repeat Command.

Sub carrier (8 bits)

This field is used to indicate the IR Sub carrier that the supporting node MUST use for repeating IR Codes.

This field MUST be expressed in kHz (kiloHertz).

Duty cycle (2 bits)

This field is used to indicate the IR Carrier that the supporting node MUST use for repeating IR Codes.

This field MUST be encoded according to Table 64.

Table 64,IR Repeater Repeat::Duty Cycle encoding

Value	Description
0x00	1/2 Duty Cycle
0x01	1/3 Duty Cycle
0x02	1/4 Duty Cycle
0x03	*Reserved*.

Values not advertised as supported in the IR Repeater Capabilities Report Command MUST be ignored by a supporting node.

Pulse time unit (12 bits)

This field is used to interpret the data blocks using Pulse Encoding type.

This field MUST be expressed in (µs) and be in the range 1..4096.

Values lower than the *Minimum Pulse Time Unit* advertised by the supporting node in the IR Repeater Capabilities Report Command MUST be ignored by a supporting node.

Values higher than the *Maximum Pulse Time Unit* advertised by the supporting node in the IR Repeater Capabilities Report Command MUST be ignored by a supporting node.

Report number (15 bits)

A sending node typically needs several report commands to describe or encode an entire IR Code. This field is used to indicate the current report number being issued. The first report MUST have this field set to 1 and incremented at every new unique report Command.

A receiving node MUST reassemble the IR Code by concatenating the data blocks of each repeat command.

Data block 1,1 of the first report MUST represent the beginning of the IR Code signal and data block M,N of the last report MUST represent the end of the IR Code signal.

Last (1 bit)

This field is used to indicate that this is the Last report.

The value 0 MUST indicate that this is not the last report and more are to follow.
The value 1 MUST indicate that this is the last report and the supporting node MUST start repeating the IR Code.

Data block type (2 bits)

This field indicates how to interpret the corresponding Data block Value field.

- 0 : bit encoding (0/1)
- 1 : pulse encoding

Data block length (6 bits)

This field indicates the length in bytes of the corresponding Data block Value field.

Data block value (6 bits)

This field MUST interpreted according to the corresponding *Data Block Type* field.

If the corresponding *Data Block Type* field is set to 0, this field MUST be interpreted as follow:

7	6	5	4	3	2	1	0
Number of bits 1							
Number of bits 2							
Data bits 1							
...							
Data bits L							

Number of bits (2 bytes)

This subfield indicates how many bits to take from the *Data bits* subfield, starting from the LSB. For example, if this subfield is set to 12, a receiving node MUST interpret the first byte and the bits 0..3 of the second byte as data bits.

Data bits (L bytes)

This subfield is used to describe successive bits to be transmitted as part of the IR Code. Each data bit (bit 0 or bit 1) MUST be repeated/interpreted according to the IR Configuration set at the supporting node. (IR Repeater Configuration Set Command)

If the corresponding Data Block Type is set to 1, this field MUST be interpreted as follow:

7	6	5	4	3	2	1	0
Number of time units 1							
...							
Number of time units P							

Number of time units (8 bits)

This subfield MUST represent the amount of time to use for an IR code pulse. The time unit is set in the *Pulse time unit* field.

This field MUST be interpreted byte-by-byte. The first byte is the first to be interpreted. It represents how long to transmit a Hi or Lo polarity.

The first byte (1) MUST represents a Hi Polarity.

If a byte of this field is in the range 0..255, it MUST be intrepreted as a multiplier of the time unit.

If a byte of this field is in the range 0..254, the polarity MUST be toggled for the next byte (from Hi to Lo or Lo to Hi)

If a byte of this field is set to 255, the polarity MUST be the same for the next byte (Hi stays Hi and Lo stays Lo)If a byte of this field is set to 0, it must ignored. The value 0 can be used for example at the first byte to start an IR Code with a Lo polarity.

4.51 Irrigation Command Class, version 1

The Irrigation Command Class provides commands to manage irrigation systems.

4.51.1 Terminology

In an irrigation system, a zone valve is turned on to allow a certain group of sprinklers to water an area (or zone) for a specified duration. The terms zone valve and valve are used synonymously, whereas a master valve is always identified as such.

A valve that is on or open lets the water run through the valve.
A valve that is off or closed blocks the water from running through the valve.

A master valve allows water to flow to a set of zone valves, and must be turned on before water flows to the zone valves. In version 1 of this Command Class, there is only one master valve in an irrigation system.

A valve is identified with a ValveID. The master valve and zone valves have their own ID pools, each starting with 1.

In some cases a master valve switch may drive a pump relay rather than an actual irrigation valve. In this case, there may be a configurable delay between the master valve (or pump) being turned on and the rest of the zone valves being turned on. When a given zone is turned on, a master valve (if present) should automatically turn on. The master valve should also automatically turn off after a device specific duration of zone valve inactivity.

A valve can be physically connected or disconnected. Attach points where valves can be connected always keep the same valve ID. A disconnected valve cannot be operated and the water does not run through a disconnected valve.

In an irrigation system, a scheduled run or run is when a set of zone valves are open and closed sequentially, one after another. Each zone valve will be turned on for a specified duration and once complete, that valve will be turned off and the next zone valve will be turned on. In a (scheduled) run, each zone valve has its own specified run duration, which is unique to that zone valve and that specific (scheduled) run. At the end of a scheduled run, the Irrigation System Shutoff Command can also be used to turn everything off.

A master valve remains on/open for the entire duration of any run, so that all zone valves will be able to provide water to their associated sprinklers.

A valve table is a list of zone valves and their corresponding run durations. A valve table can be created, configured and stored in an irrigation device. The valve table can subsequently be executed (or run) using a single command.

4.51.2 Compatibility considerations

4.51.2.1 Multi Channel considerations

CC:006B.01.00.22.001 Multi Channel End Points SHOULD NOT support the Irrigation Command Class.

4.51.3 Interoperability considerations

CC:006B.01.00.32.001 A node supporting the Irrigation Command Class SHOULD be implemented as a Multi Channel Device with the master valve and each zone valve implemented as a separate Multi Channel End Point.

CC:006B.01.00.32.002 The Root Device SHOULD support the following Command Classes:

- Irrigation Command Class
- Schedule Command Class

CC:006B.01.00.32.003 Each End Point SHOULD support the following Command Class:

- Binary Switch Command Class.

CC:006B.01.00.32.004 A node supporting optional sensors SHOULD implement each sensor as a separate Multi Channel End Point.

The Notification Command Class, version 7 or newer can be used for reporting Irrigation events/states, using the Irrigation Notification Type.

CC:006B.01.00.32.005 Scheduling can be done using the Schedule Command Class. If supported, the Schedule Command Class SHOULD support alternating days and odd/even date processing.

CC:006B.01.00.33.001
CC:006B.01.00.32.006 Some supporting nodes MAY support having only 1 zone valve open at a time. A controlling node SHOULD be aware that opening a zone valve may cause another currently open zone valve to close.

4.51.3.1 Controlling methods

A system integrator should be aware that if an irrigation control device is controlled using both End Point Binary Switch Command Class and Root Device Irrigation Command Class at the same time, this may cause unpredictable behavior. This applies whether commands are issued instantly or scheduled.

CC:006B.01.00.32.007 It is RECOMMENDED that an irrigation controlling system is managed either via direct End Point control or via the Irrigation Command Class.

An irrigation device can be driven directly from a Z-Wave controlling node, turning on the master valve and sequentially turning on each zone valve for a specified duration.

The controlling node can also use the concept of a valve table to turn on a set of valves sequentially. A valve table allows a single command, the Irrigation Valve Table Run Command, to trigger a run. The controlling node must first create the valve tables, using the Irrigation Valve Table Set Command. The run can triggered by sending the Irrigation Valve Table Run Command at the desired time. The irrigation device will sequentially run the valves in each of the specified valve tables for their associated runtimes. Depending on the irrigation system, the controlling node may need to turn on the master valve manually using the Irrigation Valve Run Command.

4.51.4 Irrigation System Info Get Command

This command is used to request a receiving node about its irrigation system information.

CC:006B.01.01.11.001 The Irrigation System Info Report Command MUST be returned in response to this command.

CC:006B.01.01.11.002 This command MUST NOT be issued via multicast addressing.

CC:006B.01.01.11.003 A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_IRRIGATION							
Command = IRRIGATION_SYSTEM_INFO_GET							

4.51.5 Irrigation System Info Report Command

This command is used to advertise irrigation system information.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_IRRIGATION							
Command = IRRIGATION_SYSTEM_INFO_REPORT							
Reserved							Master Valve
Total Number of Valves							
Total Number of Valve Tables							
Reserved				Valve Table Max Size			

Reserved

CC:006B.01.02.11.001 This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Master Valve (1 bit)

This field is used to indicate if a master valve is supported by the sending node.

CC:006B.01.02.11.002 The value 1 MUST indicate that the sending node supports a master valve.
CC:006B.01.02.11.003 The value 0 MUST indicate that the sending node does not support a master valve.

Total Number of Valves (8 bits)

This field is used to advertise the total number of zone valves supported by the device

CC:006B.01.02.11.004 The implemented Valve ID values MUST be in a sequence starting from 1, i.e. a node supporting 10 valves MUST accept Valve ID values in the range 1..10.

Total Number of Valve Tables (8 bits)

This field is used to advertise the total number of valve tables that can be created/stored in the device.

CC:006B.01.02.11.005 The implemented Valve Table ID MUST be in a sequence starting from 1, i.e. a node supporting 10 valve tables MUST accept Valve Table ID values in the range 1..10.

Valve Table Max Size (4 bits)

This field is used to advertise the maximum number of entries per valve table supported by the sending node.

CC:006B.01.02.11.006 The value MUST be in the range 1..15.

4.51.6 Irrigation System Status Get Command

This command is used to request a receiving node about its irrigation system status.

CC:006B.01.03.11.001 The Irrigation System Status Report Command MUST be returned in response to this command.

CC:006B.01.03.11.002 This command MUST NOT be issued via multicast addressing.

CC:006B.01.03.11.003 A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_IRRIGATION							
Command = IRRIGATION_SYSTEM_STATUS_GET							

4.51.7 Irrigation System Status Report Command

This command is used to advertise irrigation system status.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_IRRIGATION							
Command = IRRIGATION_SYSTEM_STATUS_REPORT							
System Voltage							
Sensor Status							
Flow Precision			Flow Scale=l/h		Flow Size		
Flow Value 1							
…							
Flow Value N							
Pressure Precision			Pressure Scale=kPa		Pressure Size		
Pressure Value 1							
…							
Pressure Value N							
Shutoff Duration							
System Error Status							
Reserved							Master Valve
Valve ID							

System Voltage (8 bits)

This field advertises the voltage level applied at the sending node.

CC:006B.01.04.11.001 The advertised value MUST be expressed in Volt. This field does not indicate the voltage type (AC or DC).

CC:006B.01.04.11.002 If the node does not have a voltage sensor measuring the voltage, this field MUST be set to 0.

Sensor Status (8 bits)

This field is used to advertise if optional sensors are currently reporting values or detecting events at the sending node.

CC:006B.01.04.11.003 This field MUST be treated as a bit mask and MUST comply with Table 65

Table 65, Irrigation System Status Report::Sensor Status encoding

Bit	Sensor Status
0	Flow Sensor currently active If this bit is set to 1, the Flow Value field value MUST advertise a valid sensor reading.
1	Pressure Sensor currently active If this bit is set to 1, the Pressure Value field value MUST advertise a valid sensor reading.
2	Rain Sensor attached and active If a rain sensor is currently attached to the device and detecting rain, this bit MUST be set to 1.
3	Moisture Sensor attached and active If a moisture sensor is currently attached to the device and detecting moisture, this bit MUST be set to 1.
4..7	*Reserved*

CC:006B.01.04.11.004 All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

Flow Precision (3 bits)

CC:006B.01.04.11.005 This field MUST indicate how many decimals are contained in the “Flow Value” field value. For example, if the “Flow Value” field is set to 1025 and the “Flow Precision” field is set to 2, the Flow Value MUST be interpreted as 10.25.

CC:006B.01.04.11.006 If the Sensor Status field’s bit 0 is set to 0, this field MUST be set to 0.

Flow Scale (2 bits)

CC:006B.01.04.11.007 This field MUST indicate which unit is used for the “Flow Value” field. This field MUST be set to 0.

CC:006B.01.04.11.008 The value 0 MUST indicate that the unit is l/h (liter/hour)

Flow Size (3 bits)

CC:006B.01.04.11.009 This field is used to advertise the length in bytes of the “Flow Value” field. This field MUST be set to 1, 2 or 4. All other values are reserved and MUST NOT be used by a sending node.

CC:006B.01.04.11.00A If the Sensor Status field’s bit 0 is set to 0, this field MUST be set to 1.

Flow Value (N bytes)

This field is used to advertise the Flow value measured by the flow sensor.

CC:006B.01.04.11.00B The length of this field in byte MUST comply with the value advertised in the Flow Size field.

CC:006B.01.04.11.00C If the Sensor Status field’s bit 0 is set to 0, this field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Pressure Precision (3 bits)

CC:006B.01.04.11.00D This field MUST indicate how many decimals are contained in the “Pressure Value” field value. For example, if the “Pressure Value” field is set to 1025 and the “Pressure Precision” field is set to 2, the Pressure Value MUST be interpreted as 10.25.

CC:006B.01.04.11.00E If the Sensor Status field’s bit 1 is set to 0, this field MUST be set to 0.

Pressure Scale (2 bits)

CC:006B.01.04.11.00F This field MUST indicate which unit is used for the “Pressure Value” field. This field MUST be set to 0.

CC:006B.01.04.11.010 The value 0 MUST indicate that the unit is kPa (kilopascal)

Pressure Size (3 bits)

CC:006B.01.04.11.011 This field is used to advertise the length in bytes of the “Pressure Value” field. This field MUST be set to 1, 2 or 4. All other values are reserved and MUST NOT be used by a sending node.

CC:006B.01.04.11.012 If the Sensor Status field’s bit 1 is set to 0, this field MUST be set to 1.

Pressure Value (N bytes)

This field is used to advertise the Pressure value measured by the pressure sensor.

CC:006B.01.04.11.013 The length of this field in bytes MUST comply with the value advertised in the Pressure Size field.

CC:006B.01.04.11.014 If the Sensor Status field’s bit 1 is set to 0, this field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Shutoff Duration (8 bits)

This field is used to indicate how many hours are left in the “shut off” mode.

For description, refer to 4.51.21 Irrigation System Shutoff Command.

System Error Status (8 bits)

This field is used to advertise if any system error is being active at the sending node.

CC:006B.01.04.11.015 This field MUST be treated as a bit mask and MUST comply with Table 66.

Table 66, System Status Report::System Error Status encoding

Bit	Description
0	The device has not been programmed
1	The device has experienced an emergency shutdown.
2	The device's pressure high threshold has been triggered.
3	The device's pressure low threshold has been triggered.
4	A valve or the master valve is reporting error. Valves can be individually checked using the Irrigation Valve Info Get Command
5..7	*Reserved*

In some cases, errors are no longer active. Any inactive errors (except valve errors) can be cleared by sending an Irrigation System Config Set Command. Refer to the Irrigation Valve Info Report Command for more details on valve specific errors.

Master Valve (1 bit)

This field is used to indicate if a master valve is currently open or closed.

CC:006B.01.04.11.016 The value 1 MUST indicate that the master valve is On / Open.

CC:006B.01.04.11.017 The value 0 MUST indicate that the master valve is Off / Closed.

Valve ID (8 bits)

This field is used to indicate the Valve ID of the first open zone valve currently On / Open.

CC:006B.01.04.11.018 If no zone valve is On / Open, this field MUST be set to 0.

CC:006B.01.04.11.019 Else, this field MUST be to a value in the range 1..{Total number of supported zone valves} by a sending node

4.51.8 Irrigation System Config Set Command

This command allows the irrigation system to be configured accordingly.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_IRRIGATION							
Command = IRRIGATION_SYSTEM_CONFIG_SET							
Master Valve Delay							
High Pressure Threshold Precision			High Pressure Threshold Scale=kPa		High Pressure Threshold Size		
High Pressure Threshold Value 1							
…							
High Pressure Threshold Value N							
Low Pressure Threshold Precision			Low Pressure Threshold Scale=kPa		Low Pressure Threshold Size		
Low Pressure Threshold Value 1							
…							
Low Pressure Threshold Value N							
Sensor Polarity							

Master Valve Delay (8 bits)

This field is used to configure a delay in seconds between turning on the master valve and turning on any “zone” valve.

CC:006B.01.05.11.001 This field MUST ignored by a receiving node if it does not implement a master valve.

CC:006B.01.05.11.002 The value MUST be expressed in seconds and the value 0 MUST indicate that no delay is applied when turning master and zone valves on.

Pressure High Threshold

These fields are used to configure the pressure high threshold at the receiving node.

CC:006B.01.05.13.001 A receiving node having no support for pressure sensor MAY ignore this field.

CC:006B.01.05.11.003 If the receiving node supports a Pressure Sensor, the node MUST issue an Irrigation System Status Report Command with the System Error Status indicating high pressure when the pressure crosses above this threshold.

CC:006B.01.05.11.004 The format of the Pressure High Threshold fields MUST comply with the format used by the Multilevel Sensor Command Class and MUST be expressed in kPa.

Pressure Low Threshold

These fields are used to configure the pressure low threshold at the receiving node.

CC:006B.01.05.13.002 A receiving node having no support for pressure sensor MAY ignore this field.

CC:006B.01.05.11.005 If the receiving node supports a Pressure Sensor, the node MUST issue an Irrigation System Status Report Command with the System Error Status indicating low pressure when the pressure crosses below this threshold.

CC:006B.01.05.11.006 The format of the Pressure Low Threshold fields MUST comply with the format used by the Multilevel Sensor Command Class and MUST be expressed in kPa.

Sensor Polarity (8 bits)

This field is used to configure optional sensors' polarity at the receiving node.

CC:006B.01.05.11.007 This field MUST be treated as a bit mask and MUST comply with Table 67. This field MUST be ignored if the 'valid' bit (bit 7) is set to 0.

Table 67, Irrigation System Config Set::Sensor Polarity encoding

Bit	Description
0	Rain Sensor Polarity (0 LOW, 1 HIGH) A receiving node having no support for rain sensor MAY ignore this field.
1	Moisture Sensor Polarity (0 LOW, 1 HIGH) A receiving node having no support for moisture sensor MAY ignore this field
2..6	*Reserved*
7	Valid This bit MUST be set to 1 to indicate that the other bits in the bitmask contain valid data.

CC:006B.01.05.11.008 All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

4.51.9 Irrigation System Config Get Command

This command is used to request a receiving node about its current irrigation system configuration.

CC:006B.01.06.11.001 The Irrigation System Config Report Command MUST be returned in response to this command.

CC:006B.01.06.11.002 This command MUST NOT be issued via multicast addressing.

CC:006B.01.06.11.003 A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_IRRIGATION							
Command = IRRIGATION_SYSTEM_CONFIG_GET							

4.51.10 Irrigation System Config Report Command

This command is used to advertise the current irrigation system configuration.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_IRRIGATION							
Command = IRRIGATION_SYSTEM_CONFIG_REPORT							
Master Valve Delay							
High Pressure Threshold Precision			High Pressure Threshold Scale=kPa		High Pressure Threshold Size		
High Pressure Threshold Value 1							
…							
High Pressure Threshold Value N							
Low Pressure Threshold Precision			Low Pressure Threshold Scale=kPa		Low Pressure Threshold Size		
Low Pressure Threshold Value 1							
…							
Low Pressure Threshold Value N							
Sensor Polarity							

CC:006B.01.07.11.001 For fields' description, refer to 4.51.8 Irrigation System Config Set Command. A sending node MUST comply with field descriptions in 4.51.8 Irrigation System Config Set Command.

4.51.11 Irrigation Valve Info Get Command

This command is used to request general information about the specified valve.

CC:006B.01.08.11.001 The Irrigation Valve Info Report Command MUST be returned in response to this command.

CC:006B.01.08.11.002 This command MUST NOT be issued via multicast addressing.

CC:006B.01.08.11.003 A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_IRRIGATION							
Command = IRRIGATION_VALVE_INFO_GET							
Reserved							Master Valve
Valve ID							

Reserved

CC:006B.01.08.11.004 This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Master Valve (1 bit)

This field is used to indicate whether the sending node requests the information of the master valve or of a zone valve.

CC:006B.01.08.11.005 The value 1 MUST indicate that the sending node requests the master valve information.

CC:006B.01.08.11.006 The value 0 MUST indicate that the sending node requests information related to the zone valve corresponding to Valve ID field.

Valve ID (8 bits)

This field is used to indicate the Valve ID if the sending node requests the information about a zone valve.

CC:006B.01.08.11.007 This field MUST be set to a Valve ID supported by the receiving node.

CC:006B.01.08.12.001 A node receiving this command for an unsupported zone valve ID SHOULD return a report for the first supported zone valve.

CC:006B.01.08.11.008 If the Master Valve field is set to 1, this field MUST be set to 1. Other values MUST be ignored by a receiving node.

4.51.12 Irrigation Valve Info Report Command

This command is used to advertise general information about a given valve.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_IRRIGATION							
Command = IRRIGATION_VALVE_INFO_REPORT							
Reserved						Conn-ected	Master Valve
Valve ID							
Nominal Current							
Valve Error Status							

Reserved

CC:006B.01.09.11.001 This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Master Valve (1 bit)

This field is used to indicate whether the sending node advertises the information of the master valve or of a zone valve.

CC:006B.01.09.11.002 The value 1 MUST indicate that the sending node advertises the master valve information.

CC:006B.01.09.11.003 The value 0 MUST indicate that the sending node advertises information related to the zone valve corresponding to Valve ID field.

Connected (1 bit)

This field indicates if the actual valve is currently connected to the node or not.

CC:006B.01.09.11.004 The value 0 MUST indicate that the valve is disconnected from the node.

CC:006B.01.09.11.005 The value 1 MUST indicate that the valve is connected to the node.

Valve ID (8 bits)

This field is used to indicate the Valve ID if the sending node requests the information about a zone valve.

CC:006B.01.09.11.006 This field MUST be set to a Valve ID supported by the receiving node.

CC:006B.01.09.11.007 If the Master Valve field is set to 1, this field MUST be set to 1. Other values MUST be ignored by a receiving node.

Nominal Current (8 bits)

This field is used to advertise the valve's nominal electric current when the valve is On / Open.

CC:006B.01.09.11.008 This field MUST be expressed as a multiple of 10mA. E.g. the value 23 represents 230 mA.

CC:006B.01.09.11.009 This field MUST be set to 0 if the Connected field is set to 0.

CC:006B.01.09.11.00A This field MUST advertise the last measured current (measured when the valve was on) when the valve is Off / Closed.

Valve Error Status (8 bits)

CC:006B.01.09.11.00B This bit mask provides valve error status fields. The field MUST be encoded according the following:

- Bit 0 indicates short circuit has been detected.
- Bit 1 indicates current high threshold has been detected.
- Bit 2 indicates current low threshold has been detected.
- Bit 3 indicates maximum flow has been detected (zone valves only).
- Bit 4 indicates flow high threshold has been detected (zone valves only).
- Bit 5 indicates flow low threshold has been detected (zone valves only).
- Bits 6-7 are reserved.

Any inactive errors can be cleared by sending an Irrigation Valve Config Set Command. Also, if the valve is turned on again, any inactive errors should be cleared as the valve is turned on.

4.51.13 Irrigation Valve Config Set Command

This command allows an irrigation valve to be configured accordingly.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_IRRIGATION							
Command = IRRIGATION_VALVE_CONFIG_SET							
Reserved							Master Valve
Valve ID							
Nominal Current High Threshold							
Nominal Current Low Threshold							
Maximum Flow Precision			Maximum Flow Scale=l/h		Maximum Flow Size		
Maximum Flow Value 1							
…							
Maximum Flow Value N							
Flow High Threshold Precision			Flow High Threshold Scale=l/h		Flow High Threshold Size		
Flow High Threshold Value 1							
…							
Flow High Threshold Value N							
Flow Low Threshold Precision			Flow Low Threshold Scale=l/h		Flow Low Threshold Size		
Flow Low Threshold Value 1							
…							
Flow Low Threshold Value N							
Sensor Usage							

Reserved

CC:006B.01.0A.11.001 This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Master Valve (1 bit)

This field is used to indicate whether the valve to configure is the master valve or a zone valve.

CC:006B.01.0A.11.002 The value 1 MUST indicate that the valve to configure is the master valve.

CC:006B.01.0A.11.003 The value 0 MUST indicate that the valve to configure is the zone valve corresponding to Valve ID field.

Valve ID (8 bits)

This field is used to indicate the Valve ID to configure.

CC:006B.01.0A.11.004 This field MUST be set to a Valve ID supported by the receiving node.

CC:006B.01.0A.11.005 If the Master Valve field is set to 1, this field MUST be set to 1. Other values MUST be ignored by a receiving node.

Nominal Current High Threshold (8 bits)

This field is used to configure the nominal current high threshold for the actual valve.

CC:006B.01.0A.11.006 The receiving node MUST issue an Irrigation System Status Report Command with the System Error Status indicating that a valve is reporting errors when the current crosses above this threshold.

CC:006B.01.0A.11.007 This field MUST be expressed as a multiple of 10mA. E.g. the value 23 represents 230 mA.

Nominal Current Low Threshold (8 bits)

This field is used to configure the nominal current low threshold for the actual valve at the receiving node.

CC:006B.01.0A.11.008 The receiving node MUST issue an Irrigation System Status Report Command with the System Error Status indicating that a valve is reporting errors when the current crosses below this threshold.

CC:006B.01.0A.11.009 This field MUST be expressed as a multiple of 10mA. E.g. the value 23 represents 230 mA.

Maximum Flow

These fields are used to configure the maximum allowed water flow for the specified valve.

CC:006B.01.0A.11.00A These fields MUST be ignored if the Master Valve field is set to 1.

CC:006B.01.0A.13.001 A receiving node having no support for flow sensor MAY ignore these fields.

CC:006B.01.0A.11.00B If the receiving node supports a flow sensor, the node MUST issue an Irrigation System Status Report Command with the System Error Status indicating that a valve is reporting errors when the water flow crosses above this threshold.

CC:006B.01.0A.11.00C This field MUST be expressed in l/h (liter/hour).

Flow High Threshold

These fields are used to configure the flow high threshold for the specified valve.

CC:006B.01.0A.11.00D These fields MUST be ignored if the Master Valve field is set to 1.

CC:006B.01.0A.13.002 A receiving node having no support for flow sensor MAY ignore these fields.

CC:006B.01.0A.11.00E If the receiving node supports a flow sensor, the node MUST issue an Irrigation System Status Report Command with the System Error Status indicating that a valve is reporting errors when the water flow crosses above this threshold.

CC:006B.01.0A.11.00F The format of the Flow High Threshold fields MUST comply with the format used by the Multilevel Sensor Command Class and MUST be expressed in l/h (liter/hour).

Flow Low Threshold

These fields are used to configure the flow low threshold for the specified valve.

CC:006B.01.0A.11.010 These fields MUST be ignored if the Master Valve field is set to 1.

CC:006B.01.0A.13.003 A receiving node having no support for flow sensor MAY ignore these fields.

CC:006B.01.0A.11.011 If the receiving node supports a flow sensor, the node MUST issue an Irrigation System Status Report Command with the System Error Status indicating that a valve is reporting errors when the water flow crosses below this threshold.

CC:006B.01.0A.11.012 The format of the Flow Low Threshold fields MUST comply with the format used by the Multilevel Sensor Command Class and MUST be expressed in l/h (liter/hour).

Sensor Usage (8 bits)

This field is used to configure if the actual valve must turn off / close when the specified sensors are active.

CC:006B.01.0A.11.013 This field MUST be ignored if the Master Valve field is set to 1.

CC:006B.01.0A.11.014 This field MUST be treated as a bit mask and MUST comply with Table 68, Irrigation Valve Config Set::Sensor Usage.

Table 68, Irrigation Valve Config Set::Sensor Usage encoding

Bit	Description
0	Use Rain Sensor The valve MUST turn off / close if rain is detected A receiving node having no support for rain sensor MAY ignore this field.
1	Use Moisture Sensor The valve MUST turn off / close if moisture is detected A receiving node having no support for moisture sensor MAY ignore this field.
2..7	*Reserved*

CC:006B.01.0A.11.015 All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

4.51.14 Irrigation Valve Config Get Command

This command is used to request the current configuration of an irrigation valve.

CC:006B.01.0B.11.001 The Irrigation Valve Config Report Command MUST be returned in response to this command.

CC:006B.01.0B.11.002 This command MUST NOT be issued via multicast addressing.

CC:006B.01.0B.11.003 A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_IRRIGATION							
Command = IRRIGATION_VALVE_CONFIG_GET							
Reserved							Master Valve
Valve ID							

Reserved

CC:006B.01.0B.11.004 This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Master Valve (1 bit)

This field is used to indicate whether the sending node requests the configuration of the master valve or of a zone valve.

CC:006B.01.0B.11.005 The value 1 MUST indicate that the sending node requests the master valve configuration.

CC:006B.01.0B.11.006 The value 0 MUST indicate that the sending node requests the configuration related to the zone valve corresponding to Valve ID field.

Valve ID (8 bits)

This field is used to indicate the Valve ID if the sending node requests the information about a zone valve.

CC:006B.01.0B.11.007 This field MUST be set to a Valve ID supported by the receiving node. A node receiving this command

CC:006B.01.0B.12.001 for an unsupported zone valve ID SHOULD return a report for the first supported zone valve.

CC:006B.01.0B.11.008 If the Master Valve field is set to 1, this field MUST be set to 1. Other values MUST be ignored by a receiving node.

4.51.15 Irrigation Valve Config Report Command

This command is used to advertise the configuration of a valve.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_IRRIGATION							
Command = IRRIGATION_VALVE_CONFIG_REPORT							
Reserved							Master Valve
Valve ID							
Nominal Current High Threshold							
Nominal Current Low Threshold							
Maximum Flow Precision			Maximum Flow Scale=l/h		Maximum Flow Size		
Maximum Flow Value 1							
…							
Maximum Flow Value N							
Flow High Threshold Precision			Flow High Threshold Scale=l/h		Flow High Threshold Size		
Flow High Threshold Value 1							
…							
Flow High Threshold Value N							
Flow Low Threshold Precision			Flow Low Threshold Scale=l/h		Flow Low Threshold Size		
Flow Low Threshold Value 1							
…							
Flow Low Threshold Value N							
Sensor Usage							

For fields' description, refer to 4.51.13 Irrigation Valve Config Set Command.

CC:006B.01.0C.11.001 A sending node MUST comply with field descriptions in 4.51.13 Irrigation Valve Config Set Command.

4.51.16 Irrigation Valve Run Command

The Irrigation Valve Run Command will run the specified valve for a specified duration.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_IRRIGATION							
Command = IRRIGATION_VALVE_RUN							
Reserved							Master Valve
Valve ID							
Duration MSB							
Duration LSB							

Reserved

CC:006B.01.0D.11.001 This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Master Valve (1 bit)

This field is used to indicate if the specified valve is the master valve or a zone valve.

CC:006B.01.0D.11.002 The value 1 MUST indicate that the specified valve is the master valve.

CC:006B.01.0D.11.003 The value 0 MUST indicate that the specified valve is the zone valve corresponding to Valve ID field.

Valve ID (8 bits)

This field is used to specify the actual Valve ID.

CC:006B.01.0D.11.004 This field MUST be set to a Valve ID supported by the receiving node.

CC:006B.01.0D.11.005 If the Master Valve field is set to 1, this field MUST be set to 1. Other values MUST be ignored by a receiving node.

Duration (16 bits)

This field is used to specify the duration of the run in seconds.

CC:006B.01.0D.11.006 The value 0 MUST indicate that the valve MUST be turned off / Closed immediately.

4.51.17 Irrigation Valve Table Set Command

This command is used to set a valve table with a list of valves and durations.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_IRRIGATION							
Command = IRRIGATION_VALVE_TABLE_SET							
Valve Table ID							
Valve ID 1							
Duration MSB 1							
Duration LSB 1							
…							
Valve ID N							
Duration MSB N							
Duration LSB N							

Valve Table ID (8 bits)

This field is used to specify the valve table ID.

CC:006B.01.0E.11.001 This field MUST be set to a Valve Table ID supported by the receiving node.

Valve ID and Duration (N * 3 bytes)

These fields are used to specify valve IDs and their associated run duration.

CC:006B.01.0E.11.002 A sending node MUST NOT specify more Valve IDs than the “Valve Table Max Size” advertised by a receiving node in the Irrigation System Info Report Command

CC:006B.01.0E.11.003 The duration MUST be expressed in seconds.

CC:006B.01.0E.13.001 This field MAY be omitted in order to erase/empty the specified valve table.

4.51.18 Irrigation Valve Table Get Command

This command is used to request the contents of the specified Valve Table ID.

CC:006B.01.0F.11.001 The Irrigation Valve Table Report Command MUST be returned in response to this command.

CC:006B.01.0F.11.002 This command MUST NOT be issued via multicast addressing.

CC:006B.01.0F.11.003 A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_IRRIGATION							
Command = IRRIGATION_VALVE_TABLE_GET							
Valve Table ID							

Valve Table ID (8 bits)

This field is used to specify the valve table ID.

CC:006B.01.0F.11.004 Valves tables MUST be identified sequentially from 1 to the total number available on the device.

4.51.19 Irrigation Valve Table Report Command

This command provides the contents of the specified Valve Table ID.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_IRRIGATION							
Command = IRRIGATION_VALVE_TABLE_REPORT							
Valve Table ID							
Valve ID 1							
Duration MSB 1							
Duration LSB 1							
…							
Valve ID N							
Duration MSB N							
Duration LSB N							

For fields' description, refer to 4.51.17 Irrigation Valve Table Set Command.

CC:006B.01.10.11.001 A sending node MUST comply with field descriptions in 4.51.17 Irrigation Valve Table Set Command.

4.51.20 Irrigation Valve Table Run Command

This command is used to run the specified valve tables sequentially.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_IRRIGATION							
Command = IRRIGATION_VALVE_TABLE_RUN							
Valve Table ID 1							
…							
Valve Table ID N							

Valve Table ID (N bytes)

This field is used to specify the list of Valve Tables to run sequentially.

CC:006B.01.11.11.001 A receiving node MUST run the indicated Valve Table ID starting by the entry 1. These fields contain a variable list of valve table IDs that will be run sequentially in the order they are listed.

4.51.21 Irrigation System Shutoff Command

This command is used to prevent any irrigation activity triggered by the Schedule CC for a specified duration.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_IRRIGATION							
Command = IRRIGATION_SYSTEM_SHUTOFF							
Duration							

Duration (8 bits)

This field is used to indicate the duration of the system shutoff.

CC:006B.01.12.11.001
CC:006B.01.12.11.002
CC:006B.01.12.11.003
Values in the range 1..254 MUST indicate how many hours the irrigation system MUST stay shut off after reception of this command. While active, the shutoff duration MUST prevent all enabled schedules (including the back-up schedule) from running. After the shutoff duration is over, any enabled schedule MUST run as configured.

CC:006B.01.12.11.004
CC:006B.01.12.11.005
The value 0 MUST indicate to turn off any running valve (including the master valve) as well as cancel any active Irrigation Valve Table Run or Schedule. Any subsequent schedule MUST run as configured.

CC:006B.01.12.11.006
The value 255 MUST indicate that the irrigation system MUST stay permanently shut off until the node receives one of the following commands:

- Irrigation System Shutoff Command with a Duration different than 255.
- Irrigation Valve Run Command
- Irrigation Valve Table Run Command

CC:006B.01.12.11.007
When permanently shut off, the node MUST NOT run any schedule

CC:006B.01.12.11.008
A receiving node MUST cancel the current shutoff and advertise a shutoff duration of 0 in the Irrigation System Status Report Command after receiving one of the following commands:

- Irrigation Valve Run Command
- Irrigation Valve Table Run Command

4.52 Language Command Class, version 1

The Language Command Class is used to specify the language settings on a device.

4.52.1 Language Set Command

This command is used to set the language at the receiving node. The receiving node SHOULD use the default language in case the received 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 (24 bits)

The code definition of the languages may 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 (16 bits)

The code definition of the countries may be found in ISO 3166-1 ‘Country Codes: Alpha-2 codes’. The use of the country field is OPTIONAL and only defined in case it is necessary for distinguishing geographical variants. The number of data fields transmitted MUST 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

4.52.2 Language Get Command

This command is used to request the current language setting in a device.

The Language Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_LANGUAGE							
Command = LANGUAGE_GET							

4.52.3 Language Report Command

This commandreturns the current language setting in a device.

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							

For fields' description, refer to 4.52.1 Language Set Command.

4.53 Lock Command Class, version 1 [DEPRECATED]

THIS COMMAND CLASS HAS BEEN DEPRECATED

A device MAY implement this command class, but it is RECOMMENDED that new implementations use the Door Lock Command Class.
If implementing this command class, it is RECOMMENDED that the Door Lock Command Class is also implemented.

The Lock Command Class is used to lock and unlock a “lock” type device, e.g. a door or window lock

4.53.1 Lock Set Command

This command is 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.

4.53.2 Lock Get Command

This command is used to request the lock state from a device.

The Lock Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_LOCK							
Command = LOCK_GET							

4.53.3 Lock Report Command

This command is used to report the lock state of a device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_LOCK							
Command = LOCK_REPORT							
Lock State							

Lock State (8 bits)

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.

4.54 Manufacturer Proprietary Command Class, version 1

The Manufacturer Proprietary Command Class is 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 Silicon Labs.

4.54.1 Manufacturer Proprietary Command

Thiscommand is used to transfer proprietary commands between devices. The Command features 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.

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 command longer than two frames, 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.

- 40 kbit/s: At least 35 ms if sending more than 2 frames back-to-back
- 100 kbit/s: At least 15 ms if sending more than 2 frames back-to-back

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. Manufacturer identifiers can be found in [12].

The first byte (Manufacturer ID 1) is the most significant byte.

Data (N bytes)

The data fields may be used for data transfer etc. The number of data fields transmitted MUST be determined from the length field in the frame.

4.55 Meter Command Class, version 1

The Meter Command Class is used to advertise instantaneous and accumulated numerical readings.

The Command Class is intended for accumulated values in physical units from a water meter or metering device (gas, electric etc.) and thereby enabling some automatic meter reading capabilities.

4.55.1 Terminology

A meter is used to monitor a resource. The meter accumulates the resource flow over time. As an option, the meter may report not only the most recent accumulated reading but also the previous reading and the time that elapsed since then. A meter may also be able to report the current resource flow. This is known as the instant value.

Meters may report two rate types. A utility installs production meters in its production facilities, while it installs consumption meters at consumer premises. The difference is the mapping of the physical flow direction to the forward direction of the meter:

- The accumulated value of a production meter grows when more resources are produced
- The accumulated value of a production meter drops when more resources are consumed

- The accumulated value of a consumption meter grows when more resources are consumed
- The accumulated value of a consumption meter drops when more resources are produced

If a meter only supports one rate type, the meter may run backwards. If a meter runs backwards, the accumulated value may eventually become negative. While the production meter of a power generator rarely runs backwards, the consumption meter in a private household with solar panels runs backwards every time the solar panels deliver more than is consumed in the household. The actual reading of a one-rate type meter reflects the net accumulated value since the meter was installed with a factory default reading of zero. The accumulated value over an arbitrary interval may be calculated by subtracting a previous accumulated value from the current accumulated value. Normal arithmetic rules ensure that this also works in case of negative values.

A production meter may be used as a consumption meter by applying a sign change to all production meter readings before using the values as consumption meter readings.

A meter device may advertise that it implements two separate registers for the production and consumption, respectively. In that case, both of these meters are always running forward: Production makes the production meter run forward while consumption makes the consumption meter run forward. Two meter reports must be used to report the values of the respective meters.

1.1.1 Meter Get Command

This command is used to request the current meter reading to a supporting node.

The Meter Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER							
Command = METER_GET (0x01)							

1.1.2 Meter Report Command

This command is used to advertise the current meter reading at the sending node.

This command MUST NOT be issued using broadcast addressing.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER							
Command = METER_REPORT (0x02)							
Meter Type							
Precision			Scale		Size		
Meter Value 1							
…							
Meter Value N							

Meter Type (8 bits)

This field is used to specify what type of metering physical unit is being reported.

This field MUST be encoded according to Table 75.

Precision (3 bits)

This field MUST indicate how many decimal places are included in the Meter Value field. For example, the *Meter Value* field set to 1025 and this field set to 2 MUST be interpreted as equal to 10.25.

The value of the precision field MUST be in the range 0..7.

Scale (2 bits)

This field MUST advertise the unit used for the *Meter Value* field.

This field MUST be encoded according to Table 76.

Size (3 bits)

This field indicates the length in bytes of the *Meter Value* field. This field MUST be set to 1, 2 or 4.

Meter Value (N bytes)

This field is used to advertise the actual meter reading.

The length of this field in bytes MUST be according to the *Size* field.

The first byte MUST be the most significant byte.

This field MUST be encoded using signed representation and comply with Table 10.

4.56 Meter Command Class, version 2

The Meter Command Class is used to advertise instantaneous and accumulated numerical readings.

The Command Class is intended for accumulated values in physical units from a water meter or metering device (gas, electric etc.) and thereby enabling some automatic meter reading capabilities.

4.56.1 Compatibility Considerations

The Meter Command Class, version 2 introduces the following functionalities:

- Commands to interview supporting nodes Meter capabilities
- New field 'Previous Meter Value' added to the Meter Report indicating consumption since previous report.
- An optional Meter Reset Command for resetting accumulated consumption.
- New meter scales have been added

Commands and fields not mentioned in this version MUST remain unchanged from version 1.

Supporting nodes MAY support several scales. A manufacturer MUST define which scale is the default scale and it MUST be described in the product manual.

4.56.2 Meter Supported Get Command

This command is used to request the supported scales in a sub meter.

The Meter Supported Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER							
Command = METER_SUPPORTED_GET (0x03)							

4.56.3 Meter Supported Report Command

This command is used to advertise the supported scales and capabilities of the sending node.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER							
Command = METER_SUPPORTED_REPORT (0x04)							
Meter Reset	Reserved		Meter Type				
Reserved				Scale Supported			

Reserved

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Meter Reset (1 bit)

This field is used to indicate if the sending node supports the Meter Reset Command.

The value 1 MUST indicate that the Meter Reset Command is supported.

The value 0 MUST indicate that the Meter Reset Command is not supported and MUST be ignored by the sending node.

Meter Type (5 bits)

This field is used to advertise the implemented meter type by the sending node.

This field MUST be encoded according to Table 75.

Scale Supported (4 bits)

This field is used to advertise the supported scales of the sending node.

The bit value ‘1’ MUST indicate support for the actual scale.
The bit value ‘0’ MUST indicate that there is no support for the actual scale.

This MUST be interpreted as a bitmask in combination with the *Meter Type* field and MUST be according to Table 74.

4.56.4 Meter Reset Command

This command is used to reset all accumulated values stored at the receiving node.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER							
Command = METER_RESET (0x05)							

A supporting node MUST reset all its supported accumulated values when receiving this command if it advertises support for Meter Reset in the Meter Supported Report Command.

A supporting node MUST ignore this command if it advertises no support for Meter Reset in the Meter Supported Report Command

4.56.5 Meter Get Command

This command is used to request the current meter reading to a supporting node .

The Meter Report Command MUST be returned in response to this command if the requested scale is supported.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER							
Command = METER_GET (0x01)							
Reserved			Scale		Reserved		

Reserved

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Scale (2 bits)

This field is used to request an actual scale for the meter reading.

This field MUST be encoded according to Table 76.

If this field is not present or set to 0x00, a supporting node MUST return a report using its default scale.

A supporting node MUST ignored this command if this field is set to a non-supported scale.

For improved compatibility with version 1 controlling nodes, a supporting node receiving this command without this field (version 1 format) SHOULD either return a version 1 Report Command or a Report Command with the *Rate Type* field set to 0x00

4.56.6 Meter Report Command

This command is used to advertise the current meter reading at the sending node.

This command MUST NOT be issued using broadcast addressing.

<table>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="8">Command Class = COMMAND_CLASS_METER</td></tr>
<tr><td colspan="8">Command = METER_REPORT (0x02)</td></tr>
<tr><td>Res</td><td colspan="2">Rate Type</td><td colspan="5">Meter Type</td></tr>
<tr><td colspan="3">Precision</td><td>Scale</td><td colspan="4">Size</td></tr>
<tr><td colspan="8">Meter Value 1</td></tr>
<tr><td colspan="8">…</td></tr>
<tr><td colspan="8">Meter Value N</td></tr>
<tr><td colspan="8">Delta Time 1</td></tr>
<tr><td colspan="8">Delta Time 2</td></tr>
<tr><td colspan="8">Previous Meter Value 1 (optional)</td></tr>
<tr><td colspan="8">…</td></tr>
<tr><td colspan="8">Previous Meter Value N (optional)</td></tr>
</table>

Fields not described below MUST remain unchanged from version 1.

Res

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Rate Type (2 bits)

This field is used to indicate if the actual reading advertises import or export values.

The Import value for a meter reading MUST indicate that the reading represents a consumed amount. The Export value for a meter reading MUST indicate that the reading represents a produced amount.

This field MUST be encoded according to Table 69.

Table 69, Meter Report::Rate Type encoding

Value	Rate Type	Version
0x00	Unspecified	1
0x01	Import (consumed)	2
0x02	Export (produced)	2
0x03	Reserved	-

Meter Type (5 bits)

This field is used to specify what type of metering physical unit is being reported.

This field MUST be encoded according to Table 75.

Precision (3 bits)

This field MUST indicate how many decimal places are included in the *Meter Value* and *Previous Meter Value* fields. For example, the *Meter Value* set to 1025 and this field set to 1 MUST be interpreted as equal to 102.5.

The value of the precision field MUST be in the range 0..7.

Scale (2 bits)

This field MUST advertise the unit used for the *Meter Value* and *Previous Meter Value* fields.

This field MUST be encoded according to Table 76.

Size (3 bits)

This field MUST indicate the length in bytes of the *Meter Value* and *Previous Meter Value* fields.

This field MUST be set to 1, 2 or 4.

Meter Value (N bytes)

This field is used to advertise the actual meter reading.

The length of this field in bytes MUST be according to the *Size* field.

The first byte MUST be the most significant byte.

The field MUST be encoded using signed representation and comply with Table 10.

Delta Time (16 bits)

This field MUST advertise the elapsed time in seconds between the readings advertised by the *Meter Value* and the *Previous Meter Value* fields. This field MUST be according to Table 70.

Table 70, Meter Report v2::Delta Time encoding

Value	Description
0x0000	No *Previous Meter Value* field is included in this command
0x0001..0xFFFE	1..65534 seconds have elapsed between reporting the *Previous Meter Value* reading and the current meter reading
0xFFFF	Unknown elapsed time between the *Previous Meter Value* reading and the current meter reading

Previous Meter Value (N bytes)

This field is used to advertise the last issued meter reading for the actual scale.

This field MUST be omitted if the *Delta Time* field is set to the value 0.
The length of this field in bytes MUST be according to the *Size* field if the *Delta Time* field is set to a non-zero value.

The length of this field in bytes MUST be according to the *Size* field.

The first byte MUST be the most significant byte.

The field MUST be encoded using signed representation and comply with Table 10.

4.57 Meter Command Class, version 3

The Meter Command Class is used to advertise instantaneous and accumulated numerical readings.

The Command Class is intended for accumulated values in physical units from a water meter or metering device (gas, electric etc.) and thereby enabling some automatic meter reading capabilities.

4.57.1 Compatibility Considerations

The Meter Command Class, version 3 introduces the following functionalities:

- New meter scales have been added.

Commands and fields not mentioned in this version MUST remain unchanged from version 2.

The manufacturer MUST define which scale is the default scale and it MUST be described in the product manual.

The default scale MUST be in the range 0..3 as version 1 and version 2 controlling nodes will not be able to interpret scale values in the range 4..7.

4.57.2 Meter Supported Report Command

This command is used to advertise the supported scales and capabilities of the sending node.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER							
Command = METER_SUPPORTED_REPORT (0x04)							
Meter Reset	Reserved		Meter Type				
Scale Supported							

Fields not described below MUST remain unchanged from version 2.

Scale Supported (8 bits)

This field is used to advertise the supported scales by the sending node.

The field MUST be treated as a bitmask in combination with the *Meter Type* field and MUST be according to Table 74.

The bit value '1' MUST indicate support for the actual scale.
The bit value '0' MUST indicate that there is no support for the actual scale.

4.57.3 Meter Get Command

This command is used to request the current meter reading to a supporting node.

The Meter Report Command MUST be returned in response to this command if the requested scale is supported.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER							
Command = METER_GET (0x01)							
Reserved			Scale			Reserved	

Reserved

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Scale (3 bits)

This field is used to request an actual scale for the meter reading.

This field MUST be encoded according to Table 76.

If this field is not present or set to 0x00, a supporting node MUST return a report using its default scale.

A supporting node MUST ignored this command if this field is set to a non-supported scale.

For improved compatibility with version 1 controlling nodes, a supporting node receiving this command without this field (version 1 format) SHOULD either return a version 1 Report Command or a Report Command with the *Rate Type* field set to 0x00

4.57.4 Meter Report Command

This command is used to advertise the current meter reading at the sending node.

This command MUST NOT be issued using broadcast addressing.

<table>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="8">Command Class = COMMAND_CLASS_METER</td></tr>
<tr><td colspan="8">Command = METER_REPORT (0x02)</td></tr>
<tr><td>Scale (2)</td><td colspan="2">Rate Type</td><td colspan="5">Meter Type</td></tr>
<tr><td colspan="3">Precision</td><td colspan="2">Scale (1:0)</td><td colspan="3">Size</td></tr>
<tr><td colspan="8">Meter Value 1</td></tr>
<tr><td colspan="8">…</td></tr>
<tr><td colspan="8">Meter Value N</td></tr>
<tr><td colspan="8">Delta Time 1</td></tr>
<tr><td colspan="8">Delta Time 2</td></tr>
<tr><td colspan="8">Previous Meter Value 1 (optional)</td></tr>
<tr><td colspan="8">…</td></tr>
<tr><td colspan="8">Previous Meter Value N (optional)</td></tr>
</table>

Fields not described below MUST remain unchanged from version 2.

Scale (3 bits)

This field MUST advertise the unit used for the *Meter Value* and *Previous Meter Value* fields.

This field is composed of two sub-fields *Scale (2)* and *Scale (1:0)* which MUST be composed and interpreted as one unit. *Scale (2)* MUST be the most significant bit.

This field MUST be encoded according to Table 76.

4.58 Meter Command Class, version 4

The Meter Command Class is used to advertise instantaneous and accumulated numerical readings.

The Command Class is intended for accumulated values in physical units from a water meter or metering device (gas, electric etc.) and thereby enabling some automatic meter reading capabilities.

4.58.1 Compatibility Considerations

The Meter Command Class, version 4 introduces the following functionalities:

- A supporting node can now support several rate types
- New meter scales have been added

Commands and fields not mentioned in this version MUST remain unchanged from version 3.

The manufacturer MUST define which scale and rate types are the default and it MUST be described in the product manual.

The default scale MUST be in the range 0..3 as version 1 and version 2 controlling nodes will not be able to interpret scale values in the range 4..7.

4.58.2 Meter Supported Report Command

This command is used to advertise the supported scales and capabilities of the sending node.

<table>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="8">Command Class = COMMAND_CLASS_METER</td></tr>
<tr><td colspan="8">Command = METER_SUPPORTED_REPORT (0x04)</td></tr>
<tr><td>Meter Reset</td><td colspan="2">Rate Type</td><td colspan="5">Meter Type</td></tr>
<tr><td>M.S.T</td><td colspan="7">Scale Supported “Byte 1”</td></tr>
<tr><td colspan="8">Number of Scale Supported Bytes to Follow (optional)</td></tr>
<tr><td colspan="8">Scale Supported “Byte 2” 1 (optional)</td></tr>
<tr><td colspan="8">…</td></tr>
<tr><td colspan="8">Scale Supported “Byte N+1” N (optional)</td></tr>
</table>

Fields not described below MUST remain unchanged from version 3.

Rate Type (2 bits)

This field is used to indicate the supported rate types by the sending node.

This field MUST be encoded according to Table 71.

Table 71, Meter Supported Report::Rate Type encoding

Value	Supported rate types	Version
0x00	Reserved	1..3
0x01	Import only (consumed)	4
0x02	Export only (produced)	4
0x03	Both Import and Export	4

Scale Supported "Byte 1" (7 bits)

This field is used to advertise the supported scales by the sending node.

The field MUST be treated as a bitmask in combination with the *Meter Type* field and MUST be according to Table 74.

The bit value '1' MUST indicate support for the actual scale.
The bit value '0' MUST indicate that there is no support for the actual scale.

M.S.T. (More Scale Types) (1 bit)

This field is used to indicate if the *Number of Scale Supported Bytes to Follow* field is included in the command.

The value 0 MUST indicate that the *Number of Scale Supported Bytes to Follow* and subsequent optional *Scale Supported* fields are omitted from the command.

The value 1 MUST indicate that the *Number of Scale Supported Bytes to Follow* field is present in the command.

Number of Scale Supported to Follow (8 bits)

This field MUST indicate the length in bytes of the subsequent *Scale Supported (bytes 2..N+1)* field.

This field MUST be omitted if the *M.S.T.* field is set to 0.

Scale Supported (bytes 2..N+1) (N bytes)

This field is used to advertise the supported scales by the sending node.

The length of this field in bytes MUST be according to the *Number of Scale Supported Bytes to Follow* field.

This field MUST be omitted if the *M.S.T.* field is set to 0 or if the *Number of Scale Supported Bytes to Follow* field is set to 0.

The field MUST be treated as a bitmask in combination with the *Meter Type* field and MUST be according to Table 74.

The bit value '1' MUST indicate support for the actual scale.
The bit value '0' MUST indicate that there is no support for the actual scale.

4.58.3 Meter Get Command

This command is used to request the current meter reading to a supporting node.

The Meter Report Command MUST be returned in response to this command if the requested scale and rate type are supported.

This command MUST NOT be issued via multicast addressing.

A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER							
Command = METER_GET (0x01)							
Rate Type		Scale			Reserved		
Scale 2							

Fields not described below MUST remain unchanged from version 3.

Rate Type (2 bits)

This field is used to request a rate type for the meter reading.

This field MUST be encoded according to Table 72.

Table 72, Meter Get::Rate Type encoding

Value	Description	Version
0x00	Default	1
0x01	Import (consumed)	4
0x02	Export (produced)	4
0x03	Reserved	-

If supported, a receiving node MUST return a report for the requested Rate Type.

If this field is not present or set to 0x00, a receiving node MUST return a Report using its default Rate Type.

Scale (3 bits)

This field is used to request an actual scale for the meter reading.

This field MUST be encoded according to Table 76.

If this field is not present or set to 0x00, a supporting node MUST return a report using its default scale.

A supporting node MUST ignored this command if this field is set to a non-supported scale.

For improved compatibility with version 1 controlling nodes, a supporting node receiving this command without this field (version 1 format) SHOULD either return a version 1 Report Command or a Report Command with the *Rate Type* field set to 0x00

The value 7 MUST indicate that the scale of the requested meter reading is advertised by the *Scale 2* field.

Scale 2 (8 bits)

This field is used to request an actual scale for the meter reading.

This field MUST be encoded according to Table 76.

This field MUST be present in the command if the *Scale* field is set to 7 (M.S.T).

This field MUST be omitted if the *Scale* field is set to a different value than 7.

4.58.4 Meter Report Command

This command is used to advertise the current meter reading at the sending node.

This command MUST NOT be issued using broadcast addressing.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER							
Command = METER_REPORT (0x02)							
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 (optional)							
…							
Previous Meter Value N (optional)							
Scale 2							

Fields not described below MUST remain unchanged from version 3.

Scale (3 bits)

This field MUST advertise the unit used for the *Meter Value* and *Previous Meter Value* fields.

This field is composed of two sub-fields *Scale (2)* and *Scale (1:0)* which MUST be composed and interpreted as one unit. *Scale (2)* MUST be the most significant bit.

This field MUST be encoded according to Table 76.

The value 7 MUST indicate that the scale of the advertised meter reading is advertised by the *Scale 2* field.

Scale 2 (8 bits)

This field is used to advertise the unit used for the *Meter Value* and *Previous Meter Value* fields.

This field MUST be present in the command if the *Scale* field is set to 7 (M.S.T).

This field MUST be omitted if the *Scale* field is set to a different value than 7.

This field MUST be encoded according to Table 73.

Table 73, Meter Report::Scale 2 encoding

Meter Type	Scale	Value	Version
Electric meter	kVar	0x00	4
	kVarh	0x01	4
	Reserved	0x02..0xFF	-
Gas meter	Reserved	0x00..0xFF	-
Water meter	Reserved	0x00..0xFF	-
Heating meter	Reserved	0x00..0xFF	-
Cooling meter	Reserved	0x00..0xFF	-

Reserved values MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

4.59 Meter Command Class, version 5

The Meter Command Class is used to advertise instantaneous and accumulated numerical readings.

The Command Class is intended for accumulated values in physical units from a water meter or metering device (gas, electric etc.) and thereby enabling some automatic meter reading capabilities.

4.59.1 Compatibility considerations

The Meter Command Class, version 5 adds support for Heating and Cooling Meter types.

The Meter Command Class, version 5 is backwards compatible with version 4.

Commands and fields not described in this version MUST remain unchanged from version 4.

4.59.2 Meter Supported Report Command

This command is used to advertise supported scales and capabilities of the sending.

<table>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="8">Command Class = COMMAND_CLASS_METER</td></tr>
<tr><td colspan="8">Command = METER_SUPPORTED_REPORT (0x04)</td></tr>
<tr><td>Meter Reset</td><td colspan="2">Rate Type</td><td colspan="5">Meter Type</td></tr>
<tr><td>M.S.T</td><td colspan="7">Scale Supported “Byte 1”</td></tr>
<tr><td colspan="8">Number of Scale Supported Bytes to Follow (Version 4 Extension)</td></tr>
<tr><td colspan="8">Scale Supported 1 “Byte 2”</td></tr>
<tr><td colspan="8">…</td></tr>
<tr><td colspan="8">Scale Supported N “Byte N+1”</td></tr>
</table>

All fields not described below MUST remain unchanged from version 4.

Table 74, Meter Supported Report::Scale Supported Bitmask encoding

Meter Type	Scale	Scale Supported Byte number	Scale Supported Bit position	Measurement mode	Version
Electric meter	kWh	Byte 1	Bit 0	Accumulated	1
	kVAh	Byte 1	Bit 1	Accumulated	2
	W	Byte 1	Bit 2	Instant	2
	Pulse count	Byte 1	Bit 3	Accumulated	2
	V	Byte 1	Bit 4	Instant	3
	A	Byte 1	Bit 5	Instant	3
	Power factor	Byte 1	Bit 6	Instant	3
	kVar	Byte 2	Bit 0	Instant	4
	kVarh	Byte 2	Bit 1	Accumulated	4
	Reserved	Byte 2	Bit 2-7	Reserved	-
Gas meter	Cubic meters	Byte 1	Bit 0	Accumulated	1
	Cubic feet	Byte 1	Bit 1	Accumulated	2
	Reserved	Byte 1	Bit 2	n/a	-
	Pulse count	Byte 1	Bit 3	Accumulated	2
	Reserved	Byte 1	Bit 4-6	Reserved	-
	Reserved	Byte 2	Bit 0-7	Reserved	-
Water meter	Cubic meters	Byte 1	Bit 0	Accumulated	1
	Cubic feet	Byte 1	Bit 1	Accumulated	1
	US gallons	Byte 1	Bit 2	Accumulated	1
	Pulse count	Byte 1	Bit 3	Accumulated	2
	Reserved	Byte 1	Bit 4-6	Reserved	-
	Reserved	Byte 2	Bit 0-7	Reserved	-
Heating meter	kWh	Byte 1	Bit 0	Accumlated	5
	Reserved	Byte 1	Bit 1-6	Reserved	-
	Reserved	Byte 2	Bit 0-7	Reserved	-
Cooling meter	kWh	Byte 1	Bit 0	Accumlated	5
	Reserved	Byte 1	Bit 1-6	Reserved	-
	Reserved	Byte 2	Bit 0-7	Reserved	-

4.59.3 Meter Report Command

This command is used to advertise the current meter reading at the sending node.

This command MUST NOT be issued using broadcast addressing.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER							
Command = METER_REPORT (0x02)							
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 (optional)							
...							
Previous Meter Value N (optional)							
Scale 2							

Fields not described below MUST remain unchanged from version 4.

Meter Type (5 bits)

This field is used to specify what type of metering physical unit is being reported.

This field MUST be encoded according to Table 75.

Table 75,Meter Report::Meter Type encoding

Value	Meter Type	Version
0x01	Electric meter	1
0x02	Gas meter	1
0x03	Water meter	1
0x04	Heating meter	5
0x05	Cooling meter	5

All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

Scale (3 bits)

This field MUST advertise the unit used for the *Meter Value* and *Previous Meter Value* fields . This field is composed of two sub-fields *Scale (2)* and *Scale (1:0),* which MUST be composed and interpreted as one unit. *Scale (2)* MUST be the most significant bit.

This field MUST be encoded according to Table 76.

The value 0x07 MUST indicate that the scale of the advertised meter reading is advertised through the *Scale 2* field.

Table 76, Meter Report::Scale encoding

Meter Type	Scale	Value	Version
Electric meter	kWh	0x00	1
	kVAh	0x01	2
	W	0x02	2
	Pulse count	0x03	2
	V	0x04	3
	A	0x05	3
	Power Factor	0x06	3
	M.S.T	0x07	4
Gas meter	Cubic meters	0x00	1
	Cubic feet	0x01	2
	Reserved	0x02	-
	Pulse count	0x03	2
	Reserved	0x04..0x06	-
	M.S.T	0x07	4
Water meter	Cubic meters	0x00	1
	Cubic feet	0x01	1
	US gallons	0x02	1
	Pulse count	0x03	2
	Reserved	0x04..0x06	-
	M.S.T	0x07	4
Heating meter	kWh	0x00	5
Cooling meter	kWh	0x00	5

4.60 Meter Command Class, version 6

The Meter Command Class is used to advertise instantaneous and accumulated numerical readings.

The Command Class is intended for accumulated values in physical units from a water meter or metering device (gas, electric etc.) and thereby enabling some automatic meter reading capabilities.

4.60.1 Compatibility Considerations

The Meter Command Class, version 6 adds a functionality to reset the accumulated value stored at receiving node to specific.values.

The Meter Command Class, version 6 is backwards compatible with version 5.

Commands and fields not described in this version MUST remain unchanged from version 5.

4.60.2 Meter Reset Command

This command is used to reset the accumulated values stored at the receiving node.

If the node advertises support for Meter Reset in the Meter Supported Report Command, the supporting node MUST reset the accumulated values to specific values, instead of setting to zero, when receiving this command.

A supporting node MUST ignore this command if it advertises no support for Meter Reset in the Meter Supported Report Command.

When a supporting node resets the value, it MUST use the scale that was used in the last issued Meter Report Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER							
Command = METER_RESET (0x05)							
Size			Meter Type				
Meter Value 1							
…							
Meter Value N							

Size (3 bits)

This field MUST indicate the length in bytes of the *Meter Value* field.
This field MUST be set to 1, 2 or 4.

Meter Type (5 bits)

This field is used to specify which Meter type values are being reseted.

This field MUST be encoded according to Table 75.

Meter Value (N bytes)

This field is used to advertise the actual meter values to set.

The length of this field in bytes MUST be according to the *Size* field.
The first byte MUST be the most significant byte.
The field MUST be encoded using signed representation and comply with Table 10.

4.61 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 optionally 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). Refer to [14] for more details about Z-Wave security.

4.61.1 Meter Table Point Adm Number Set Command

This 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

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Number of Meter Point Adm Number Characters (5 bits)

Number of characters in the meter point administration number(1…32).

Meter Point Adm Number Character (N bytes)

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).

4.62 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

4.62.1 Meter Table Point Adm. Number Get Command

This command is used to request the Meter Point Administration Number to identify customer.

The Meter Table Adm. Number Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.

A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER_TBL_MONITOR							
Command = METER_TBL_TABLE_POINT_ADM_NO_GET							

4.62.2 Meter Table Point Adm. Number Report Command

This command reports 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_POINT_ADM_NO_REPORT							
Reserved			Number of Meter Point Adm. Number Characters				
Meter Point Adm. Number Character 1							
…							
Meter Point Adm. Number Character N							

Fields MUST be according to descriptions in the Meter Table Point Adm Number Set Command from the Meter Table Configuration Command Class, version 1.

4.62.3 Meter Table ID Get Command

This command is used to request the parameters used for identification of customer and metering device.

The Meter Table ID Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.

A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER_TBL_MONITOR							
Command = METER_TBL_TABLE_ID_GET							

4.62.4 Meter Table ID Report Command

This command reports 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_REPORT							
Reserved			Number of Meter ID Characters				
Meter ID Character 1							
…							
Meter ID Character N							

Reserved

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Number of Meter ID Characters (5 bits)

This field is used to indicate the length of the *Meter ID Character* field in bytes.

This field MUST be in the range 1..32.

Meter ID Character (N bytes)

This field is used to identify the individual metering device.

The length of this field in bytes MUST be according to the *Number of Meter ID Characters* field.

Each byte of this field MUST be encoded with ASCII representation and be in the range 0x00..0x7F.

A controlling node can use the Manufacturer Specific Command Class in conjunction for product identification.

4.62.5 Meter Table Capability Get Command

This command is used to request the meter table capabilities of a supporting device.

The Meter Table Capability Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.

A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER_TBL_MONITOR							
Command = METER_TBL_TABLE_CAPABILITY_GET							

4.62.6 Meter Table Capability Report Command

This command is used to advertise meter table capabilities of the sending node.

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 bits)

This field is used to indicate if the actual reading advertises import or export values.

The Import value for a meter reading MUST indicate that the reading indicates a consumed amount.

The Export value for a meter reading MUST indicate that the reading indicates a produced amount.

This field MUST be encoded according to the Rate Types values defined in [21].

Meter Type (6 bits)

This field is used to specify the type of metering physical unit that is being reported.

This field MUST be encoded according to the Meter Types values defined in [21].

Reserved

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Pay Meter (4 bits)

This field is used to indicate the type of payment that is used for the meter.

This field MUST be encoded according to the Pay meter values defined in [21].

Dataset Supported (24 bits)

This field is used to advertise which datasets are supported and can be requested from the metering node with the Meter Table Current Data Get Command.

This field MUST be encoded according to the Meter dataset bitmask values defined in [21].

Dataset History Supported (24 bits)

This field is used to advertise which datasets are supported and can be requests from the metering node with the Meter Table Historical Data Get Command.

This field MUST be encoded according to the Meter dataset bitmask values defined in [21].

Data History Supported (24 bits)

This field is used to advertise the total number of entries that the supporting node can hold in memory for historical values.

This field MUST be in the range 0..16777215.
The value 0 MUST indicate that Historical data cannot be requested to the sending node.

4.62.7 Meter Table Status Supported Get Command

This command is used to request the supported operating status event parameters and logging depth of the metering device.

The Meter Table Status Supported Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.

A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER_TBL_MONITOR							
Command = METER_TBL_STATUS_SUPPORTED_GET							

4.62.8 Meter Table Status Supported Report Command

This 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 (24 bits)

This field is used to advertise which operating statuses are supported and can be sent by the metering node in the Meter Table Status Report Command.

This field MUST be encoded as a bitmask according to the Operating status values defined in [21].

If the Operating Status event is supported, the corresponding bit MUST be set to 1.

If the Operating Status event is not supported, the corresponding bit MUST be set to 0.

A supporting node MAY support no operating status event and set this field to 0.

Status Event Log Depth (8 bits)

This field MUST indicate the supported depth of the event log.

This field MUST be in the range 0..255.

If the supporting node can only return the current status, this field MUST be set to 0.

4.62.9 Meter Table Status Depth Get Command

This 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.

The Meter Table Status Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.

A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

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 bits)

This field is used to indicate the number of last recorded events that should be returned.

The value 0x00 MUST indicate that the receiving node MUST return the current status only.

Values in the range 0x01..0xFE MUST indicate the number of last recorded event that the receiving node SHOULD return.

The value 0xFF MUST indicate that the receiving node SHOULD return its entire status event log.

4.62.10 Meter Table Status Date Get Command

This command is used to request a number of status events recorded in a certain time interval.

The Meter Table Status Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

If a supporting node does not support a status event log history, it MUST return the current state of the meter (refer to the 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 bits)

This field is used to indicate the maximum number of Meter Table Status Report Command that can be returned to advertise the requested status event history. The most recent recorded log entries MUST be returned first.

The value 0x00 MUST indicate that there is no maximum number of reports for returning the event log and the supporting node MUST return as many Reports as necessary to advertise the requested entry logs.

Values in the range 0x01..0xFF MUST indicate an actual upper limit number of reports to advertise the requested entry logs.

Start/Stop - Year (16 bits)

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

Start/Stop - Month (8 bits)

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

Start/Stop - Day (8 bits)

Specify the day of the month between 01 and 31.

Start/Stop - Hour Local Time (8 bits)

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

Start/Stop - 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.

Start/Stop - 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.

4.62.11 Meter Table Status Report Command

This command is used to advertise the current status and optionally historical status data of the meter.

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 bits)

This field MUST be used if multiple Meter Table Status Report Commands are used to report the requested Operating status.

Values in the range 0x00..0xFE MUST indicate the actual number of commands following the actual command.

The value 0xFF MUST indicate that the number of remaining commands have not been calculated yet or is higher than 255.

Current Operating Status (24 bits)

This field is used to advertise which operating statuses are currently active at the sending node.

This field MUST be encoded as a bitmask according to the Operating status values defined in [21].

If the Operating Status event is active, the corresponding bit MUST be set to 1.

If the Operating Status event is not active, the corresponding bit MUST be set to 0.

Event – Type (N * 1 bit)

This field is used to advertise the type of the actual event in the event log. It represents the transition to or from a state.

The value 0 MUST indicate that the sending node entered the state described by the corresponding *Operating Status Event ID* at the reported time.

The value 1 MUST indicate that the sending node left the state described by the corresponding *Operating Status Event ID* at the reported time.

Event – Operating Status Event ID (N * 5 bits)

This field is used to advertise the identifier of the actual event in the event log.

This field MUST be encoded according to the Operating status Event identifier values defined in [21].

Event - Year (N * 16 bits)

This field is used to specify the year in the usual Gregorian calendar for the actual event. The first byte (Year 1) MUST be the most significant byte.

Event - Month (N * 8 bits)

This field is used to specify the month of the year between 01 (January) and 12 (December) for the actual event. This field MUST be in the range 1..12.

Event - Day (N * 8 bits)

This field is used to specify the day of the month for the actual event. This field MUST be in the range 1..31

Event - Hour Local Time (N * 8 bits)

This field is used to specify the number of complete hours that have passed since midnight in local time for the actual event. This field MUST be in the range 0..23.

Event - Minute Local Time (N * 8 bits)

This field is used to specify the number of complete minutes that have passed since the start of the hour in local time for the actual event. This field MUST be in the range 0..59.

Event - Second Local Time (N * 8 bits)

This field is used to specify the number of complete seconds since the start of the minute in local time for the actual event. The value 60 used to keep UTC from wandering away is not supported. This field MUST be in the range 0..59.

4.62.12 Meter Table Current Data Get Command

This command is used to request a number of time stamped values (current) in physical units according to the dataset mask.

The Meter Table Current Data Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.

A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

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 (24 bits)

This field is used to indicate which datasets are requested from the supporting node.

This field MUST be encoded as a bitmask and according to the Meter dataset bitmask values defined in [21].

4.62.13 Meter Table Current Data Report Command

This command is is used to report a number of time stamped values.

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							

Reserved

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Reports to follow (8 bits)

This field MUST be used if multiple Meter Table Current Data Report Commands are used to report the requested values.

Values in the range 0x00..0xFE MUST indicate the actual number of commands following the actual command.

The value 0xFF MUST indicate that the number of remaining commands have not been calculated yet or is higher than 255.

Rate Type (2 bits)

This field is used to specify the type of parameters advertised in this command. This field MUST be encoded according to the rate types values defined in [21].

Dataset (24 bits)

This field is used to indicate which datasets are included in this command. This field MUST be encoded as a bitmask and according to the Meter dataset bitmask values defined in [21].

The length of the *Current Meter Precision*, *Current Meter Scale* and *Current Value* fields MUST be according to the number of bits set in this field.

Year (16 bits)

This field is used to specify the year in the usual Gregorian calendar. The first byte (Year 1) MUST be the most significant byte. A year equal to 0x0000 MUST indicate that an accumulated value is not determined yet.

Month (8 bits)

This field is used to specify the month of the year between 01 (January) and 12 (December). This field MUST be in the range 1..12.

Day (8 bits)

This field is used to specify the day of the month. This field MUST be in the range 1..31.

Hour Local Time (8 bits)

This field is used to specify the number of complete hours that have passed since midnight in local time. This field MUST be in the range 0..23.

Minute Local Time (8 bits)

This field is used to specify the number of complete minutes that have passed since the start of the hour in local time. This field MUST be in the range 0..59.

Second Local Time (8 bits)

This field is used to specify the number of complete seconds since the start of the minute in local time. The value 60 used to keep UTC from wandering away is not supported. This field MUST be in the range 0..59.

Meter Precision (N * 3 bits)

This field is used to indicate how many decimal places are included in the corresponding *Current Value* field. For example, the *Current Value* field set to 1025 with this field set to 2 MUST be interpreted as equal to 10.25.

Meter Scale (N * 5 bits)

This field is used to indicate the scale (unit) for the corresponding *Current Value* field reading. This field MUST be encoded according to the Meter Scale values defined in [21].

Current Value (N * 4 bytes)

This field is used to advertise an actual meter reading.

The first byte MUST be the most significant byte.

This field MUST be encoded using signed representation and comply with Table 10, Signed field encoding (two's complement representation).

Readings MUST be advertised according to the *Dataset* field. The first reading MUST correspond to the first bit set in the *Dataset* field, the second reading MUST correspond to the second bit set in the *Dataset* field, and so on.

A controlling node MUST always show the value even if the reading type or scale are unknown.

A controlling node SHOULD implement the capability to update its list of Meter datasets and scales, so that new Meter readings and scales added in [21] are not presented as unknown. If a controlling node receives an unknown Meter reading or scale, it SHOULD allow the user to assign a free-text description to that Meter reading

4.62.14 Meter Table Historical Data Get Command

This command is used to request a number of time stamped values (historical) in physical units according to rate type, dataset bitmask and time interval.

The Meter Table Historical Data Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.

A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

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 bits)

This field is used to indicate the maximum number of Meter Table Historical Data Report Commands that can be returned to advertise the requested historical data. The most recent recorded values MUST be returned first.

The value 0x00 MUST indicate that there is no maximum number of reports for returning the event log and the supporting node MUST return as many Reports as necessary to advertise the requested historical values.

Values in the range 0x01..0xFF MUST indicate an actual upper limit number of reports to advertise the requested historical values.

Dataset History (24 bits)

This field is used to indicate which datasets are requested from the supporting node.

This field MUST be encoded as a bitmask and according to the Meter dataset bitmask values defined in [21].

Start/Stop Year (16 bits)

This field is used to specify the year in the usual Gregorian calendar. The first byte (Year 1) MUST be the most significant byte.

Start/Stop Month (8 bits)

This field is used to specify the month of the year between 01 (January) and 12 (December). This field MUST be in the range 1..12.

Start/Stop Day (8 bits)

This field is used to specify the day of the month. This field MUST be in the range 1..31.

Start/Stop Hour Local Time (8 bits)

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

Start/Stop Minute Local Time (8 bits)

This field is used to specify the number of complete minutes that have passed since the start of the hour (00-59) in local time. This field MUST be in the range 0..59.

Start/Stop Second Local Time (8 bits)

This field is used to specify the number of complete seconds since the start of the minute in local time. The value 60 used to keep UTC from wandering away is not supported. This field MUST be in the range 0..59.

4.62.15 Meter Table Historical Data Report Command

This command is used to report a number of time stamped values.

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 bits)

This field MUST be used if multiple Meter Table Current Data Report Commands are used to report the requested values.

Values in the range 0x00..0xFE MUST indicate the actual number of commands following the actual command.

The value 0xFF MUST indicate that the number of remaining commands have not been calculated yet or is higher than 255.

Reserved

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Rate Type (2 bits)

This field is used to specify the type of parameters advertised in this command. This field MUST be encoded according to the rate types values defined in [21].

Dataset (24 bits)

This field is used to indicate which datasets are included in this command. This field MUST be encoded as a bitmask and according to the Meter dataset bitmask values defined in [21].

If no historical data has been registered for the requested period, this field MUST be set to 0 and the *Historical Scale*, *Historical Precision* and *Historical Value* fields MUST be omitted from the command.

Historical Year (16 bits)

This field is used to specify for the dataset the year in the usual Gregorian calendar. The first byte (Year 1) MUST be the most significant byte. A year equal to 0x0000 indicates that an accumulated value is not determined yet.

Historical Month (8 bits)

This field is used to specify for the dataset the month of the year between 01 (January) and 12 (December). This field MUST be in the range 1..12.

Historical Day (8 bits)

This field is used to specify for the dataset the day of the month. This field MUST be in the range 1..31.

Historical Hour Local Time (8 bits)

This field is used to specify for the dataset the number of complete hours that have passed since midnight in local time. This field MUST be in the range 0..23.

Historical Minute Local Time (8 bits)

This field is used to specify for the dataset the number of complete minutes that have passed since the start of the hour in local time. This field MUST be in the range 0..59.

Historical Second Local Time (8 bits)

This field is used to specify for the dataset the number of complete seconds since the start of the minute in local time. The value 60 used to keep UTC from wandering away is not supported. This field MUST be in the range 0..59.

Historical Precision (N * 3 bits)

This field is used to specify how many decimal places are included in the corresponding *Historical Value* field. For example, the *Historical Value* field set to 1025 with this field set to 2 MUST be interpreted as equal to 10.25.

Historical Scale (5 bits)

This field is used to indicate the scale (unit) of the corresponding *Historical Value* field reading. This field MUST be encoded according to the Meter Scale values defined in [21].

Historical Value

This field is used to advertise an actual historical meter reading.

The first byte MUST be the most significant byte.

This field MUST be encoded using signed representation and comply with Table 10, Signed field encoding (two's complement representation).

Readings MUST be advertised according to the *Dataset* field. The first reading MUST correspond to the first bit set in the *Dataset* field, the second reading MUST correspond to the second bit set in the *Dataset* field, and so on.

A controlling node MUST always show the value even if the reading type or scale are unknown.

A controlling node SHOULD implement the capability to update its list of Meter datasets and scales, so that new Meter readings and scales added in [21] are not presented as unknown. If a controlling node receives an unknown Meter reading or scale, it SHOULD allow the user to assign a free-text description to that Meter reading

4.63 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.

4.63.1 Compatibility Considerations

The Meter Table Monitor Command Class, version 2 is backwards compatible with the Meter Table Monitor Command Class, version 1.

All commands and fields not mentioned in this version MUST remain unchanged from the Meter Table Monitor Command Class, version 1.

4.63.2 Meter Table Point Adm. Number Report Command

This command reports 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_POINT_ADM_NO_REPORT							
Reserved			Number of Meter Point Adm. Number Characters				
Meter Point Adm. Number Character 1							
…							
Meter Point Adm. Number Character N							

Fields not described below MUST remain unchanged from version 1.

Number of Meter Point Adm. Number Characters (5 bits)

If the *Meter Table Point Adm, Number* has not been set using the Meter Table Configuration Command Class, this field MUST be set to 0x00 and the *Meter Point Adm. Numbers Character* field MUST be omitted.

4.63.3 Meter Table ID Report Command

This command reports 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_REPORT							
Reserved			Number of Meter ID Characters				
Meter ID Character 1							
…							
Meter ID Character N							

Fields not described below MUST remain unchanged from version 1.

Number of Meter ID Characters (5 bits)

This field is used to indicate the length of the *Meter ID Character* field in bytes.

This field MUST be in the range 0..32.

The value 0 MUST indicate that the Meter ID is not used and the *Meter ID Character* field MUST be omitted from the command.

4.63.4 Meter Table Capability Report Command

This command is used to advertise meter table capabilities.

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							

Fields not described below MUST remain unchanged from version 1.

Rate Type (2 bits)

This field is used to indicate if the actual reading advertises import or export values.

The Import value for a meter reading MUST indicate that the reading indicates a consumed amount. The Export value for a meter reading MUST indicate that the reading indicates a produced amount.

This field MUST be encoded according tothe Rate Types values defined in [21].

If the *Meter Type* field is set to Submeter (0x0B), this field MUST be set to Import (0x01).

4.63.5 Meter Table Current Data Report Command

This command is used to report a number of time stamped values.

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							

Fields not described below MUST remain unchanged from version 1.

Operating Status Indication (1 bit)

This field is used to indicate that the reported meter data is measured while the meter is in an operating status different from Normal (0x00) e.g. accuracy warning or clock not accurate.

The value 1 MUST indicate that the meter is operating in a status different from Normal

The value 0 MUST indicate that the meter is operating in Normal mode (Operating status=0x00).

Rate Type (2 bits)

This field is used to indicate the type of parameters in the report. This field MUST be encoded according to the Rate Types values defined in [21].

If the *Meter Type* field is set to Submeter (0x0B), this field MUST be set to Import (0x01).

4.63.6 Meter Table Historical Data Report Command

This command is used to report a number of time stamped values.

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							

Fields not described below MUST remain unchanged from version 1.

Operating Status Indication (1 bit)

This field is used to indicate that the reported meter data is measured while the meter is in an operating status different from Normal (0x00) e.g. accuracy warning or clock not accurate.

The value 1 MUST indicate that the meter is operating in a status different from Normal

The value 0 MUST indicate that the meter is operating in Normal mode (Operating status=0x00).

Rate Type (2 bits)

This field is used to indicate the type of parameters in the report. This field MUST be encoded according to the Rate Types values defined in [21].

If the *Meter Type* field is set to Submeter (0x0B), this field MUST be set to Import (0x01).

4.64 Meter Table Monitor Command Class, version 3

The Meter Table Monitor Command Class, version 3 is used for advanced metering applications. It allows reading 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

4.64.1 Compatibility Considerations

The Meter Table Monitor Command Class, version 3 is backwards compatible with the Meter Table Monitor Command Class, version 2.

All commands and fields not mentioned in this version MUST remain unchanged from the Meter Table Monitor Command Class, version 2.

The list of supported Meter Types, Meter scales and Meter dataset bitmasks is moved to [21]. Values not defined in [21] are reserved and MUST NOT be used by a supporting node.

4.65 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 may be configured to return different data sets at different intervals using both the primary and secondary push commands.

4.65.1 Meter Table Push Configuration Set Command

This command is used to request the meter to send a Current Data Report at a given interval. The meter may be configured to return different data sets 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 bits)

This field is used to configure if the Meter Table Status Report Command (refer to 4.62.11) 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 (24 bits)

The Push Dataset parameter is use to indicate which parameters are requested to be pushed from the meter. This field MUST be encoded according to the Meter Dataset defined in [21].

Interval Months (8 bits)

Specify the number of months between pushing the push dataset.

Interval Day (8 bits)

Specify the number of days between pushing the push dataset.

Interval Hours (8 bits)

Specify the number of hours between pushing the push dataset.

Interval Minute (8 bits)

Specify the number of minutes between pushing the push dataset.

Push Node ID (8 bits)

Specify the node ID of the node to receive push dataset in the given interval.

4.65.2 Meter Table Push Configuration Get Command

This command is used to request the meters push configuration

The Meter Table Push Configuration Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER_TBL_PUSH							
Command = METER_TBL_PUSH_CONFIGURATION_GET							

4.65.3 Meter Table Push Configuration Report Command

This command is used report the current Push Configuration

7	6	5	4	3	2	1	0
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							

Refer to Meter Table Push Configuration Set Command (section 4.65.1) for detailed description of the fields.

4.66 Move To Position Window Covering Command Class, version 1 [OBSOLETED]

THIS COMMAND CLASS HAS BEEN OBSOLETED

New implementations MUST NOT support this command class.

Window covering device implementations SHOULD support the Window Covering Command Class.

The Move To Position Window Covering Command Class is used to control the position of a window covering device.

The Move To Position Window Covering Command Class is an actuator control command class. Refer to 3.6.

4.66.1 Move To Position Set Command

This command is used to instruct a window covering to move to a new position.

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 encoding of the Value field MUST be according to Table 77.

Table 77, Move To Position Set :: Value

Value		Level	State
0	(0x00)	0%	Closed
1..99	(0x01..0x63)	Almost closed .. 100% Open	Open
…		*reserved*	*reserved*
255	(0xFF)	100% Open	Open

All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

4.66.2 Move To Position Get Command

This command is used to request the status of a window covering device.

The Move To Position Report command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_MTP_WINDOW_COVERING							
Command = MOVE_TO_POSITION_GET							

4.66.3 Move To Position Report Command

This command is used to advertise the status of a window covering device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_MTP_WINDOW_COVERING							
Command = MOVE_TO_POSITION_REPORT							
Value							

Value (8 bits)

The encoding of the Value field MUST be according to Table 77.

The Value field SHOULD advertise the current value of the device hardware; also while in transition to a new target value.

A controlling device MUST NOT assume that the Value is identical to a value previously issued with a Set command when a transition has ended.

4.67 Multilevel Sensor Command Class, version 1-4

The Multilevel Sensor Command Class is used to advertise numerical sensor readings..

4.67.1 Multilevel Sensor Get Command

This command is used to request the current reading from a multilevel sensor.

CC:0031.01.04.11.001 The Multilevel Sensor Report Command MUST be returned in response to this command.

CC:0031.01.04.11.002 This command MUST NOT be issued via multicast addressing.

CC:0031.01.04.11.003 A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SENSOR_MULTILEVEL							
Command = SENSOR_MULTILEVEL_GET							

4.67.2 Multilevel Sensor Report Command

This command is used by a supporting node to advertise its current sensor reading for its supported sensor type.

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							

Sensor Type (8 bits)

This field is used to specify the sensor type of the actual sensor reading.

CC:0031.01.05.11.001 This field MUST be set to a value defined in [17]. Values not defined in [17] are reserved and MUST NOT be used.

CC:0031.01.05.11.002 A node MUST support as a minimum the highest Multilevel Sensor Command Class version associated with the Scale and Type it supports. The minimum required version for each Scale and Type is specified in [17].

Precision (3 bits)

CC:0031.01.05.11.003 This field is used to indicate how many decimal places are included the Sensor Value field. For example, the Sensor Value 1025 with precision 2 MUST be interpreted as equal to 10.25.

Scale (2 bits)

This field is used to indicate what scale is used for the actual sensor reading.

CC:0031.01.05.11.004 This field MUST be set to a value defined in [17]. Values not defined in [17] are reserved and MUST NOT be used.

CC:0031.01.05.11.005 A node MUST implement as a minimum the highest Multilevel Sensor Command Class version associated with the Scale and Type it supports. The minimum required version for each Scale and Type is specified in [17].

Size (3 bits)

This field is used to indicate the length in bytes of the Sensor Value field.

CC:0031.01.05.11.006 This field MUST be set to 1, 2 or 4.

Sensor Value (N bytes)

This field is used to advertise the value of the actual sensor reading.

CC:0031.01.05.11.007 The length of this field in bytes MUST be according to the Size field value.

CC:0031.01.05.11.008 The first byte MUST be the most significant byte.

CC:0031.01.05.11.009 This field MUST be encoded using signed representation and comply with Table 10, Signed field encoding (two's complement representation).

CC:0031.01.05.11.00A A controlling node receiving this command MUST always show the sensor value as is even though the Sensor Type and/or Scale are unknown.

CC:0031.01.05.12.001 A controlling node SHOULD implement the capability to update its list of Sensor Type and Scales, so that new Sensor Types and Scales added in [17] are not presented as unknown. If a controlling node receives an unknown Sensor Type or Scale, it SHOULD allow the user to assign a free-text description to the sensor reading.

CC:0031.01.05.12.002

4.68 Multilevel Sensor Command Class, Version 5-11

The Multilevel Sensor Command Class is used to advertise numerical sensor readings .

4.68.1 Compatibility considerations

A node supporting version 5 or newer may support several sensor types and advertise the readings for each of them.

Version 5 of this command class is extended with the following functionality:

- 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

Version 6 of this command class is extended with the following functionality:

- Additional sensor types

Version 7 of this command class is extended with the following functionality:

- Additional sensor types

Version 8 of this command class is extended with the following functionality:

- New Sensor Types and Scale Values for rotation and linear movement.
- New Sensor Types and Scale Values for Smoke Density

Multilevel Sensor Command Class, version 8 deprecates the Sensor Type "Angle Position".

Version 9 of this command class is extended with the following functionality:

- New Sensor Types and Scale Values for Water Flow and Water Pressure
- New Sensor Types and Scale Values for RF Signal Strength

Version 10 of this command class is extended with the following functionality:

- New Sensor Types and Scale Values for Particulate Matter 10 and Respiratory rate
- New Scale Values for CO and VOC Sensors Types

Version 11 of this command class is extended with the following functionality:

- New Sensor Types and Scale Values.
- Moved the list of assigned Sensor Types and Scale Values to an external registry [17].
- Deprecated the “General Purpose” Sensor Type

4.68.1.1 Unknown Multilevel Sensor Types and Scales

CC:0031.05.05.22.001 A controlling node SHOULD implement the capability to update its Multilevel sensor types and scales list, so that new Types and Scales added in [17] are not presented as unknown. If a controlling node

CC:0031.05.05.22.002 receives an unknown Type or Scale, it SHOULD allow the user to assign a free-text description to that Type or Scale.

4.68.2 Multilevel Sensor Get Supported Sensor Command

This command is used to request the supported Sensor Types from a supporting node.

CC:0031.05.01.11.001 The Multilevel Sensor Supported Sensor Report Command MUST be returned in response to this command.

CC:0031.05.01.11.002 This command MUST NOT be issued via multicast addressing.

CC:0031.05.01.11.003 A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SENSOR_MULTILEVEL							
Command = SENSOR_MULTILEVEL_SUPPORTED_GET_SENSOR							

4.68.3 Multilevel Sensor Supported Sensor Report Command

This command is used to advertise the supported Sensor Types by a supporting node.

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 (N bytes)

This field is used to advertise the supported sensor types by sending node.

CC:0031.05.02.11.001 This field MUST be treated as a bit mask and interpreted as follows:

- Bit 0 in Bit Mask 1 indicates if Sensor Type = Air Temperature (0x01) is supported
- Bit 1 in Bit Mask 1 indicates if Sensor Type = General Purpose (0x02) is supported
- Bit 2 in Bit Mask 1 indicates if Sensor Type = luminance (0x03) is supported
- …

The list of Sensor Types is defined in [17].

CC:0031.05.02.11.002 The value 1 MUST indicate that the corresponding Sensor Type is supported.

The value 0 MUST indicate that the corresponding Sensor Type is not supported.

CC:0031.05.02.11.003 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 MUST be determined from the length field in the frame.

Note that the mapping of bit 0 to Sensor Type = 1 differs from the support mapping used by the Notification Command Class. The Notification Command Class maps bit 1 to Notification Type =1.

4.68.4 Multilevel Sensor Get Supported Scale Command

This command is used to retrieve the supported scales of the specific sensor type from the Multilevel Sensor device.

CC:0031.05.03.11.001 The Multilevel Sensor Supported Scale Report Command MUST be returned in response to this command.

CC:0031.05.03.11.002 This command MUST NOT be issued via multicast addressing.

CC:0031.05.03.11.003 A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

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 bits)

This field is used to request a given sensor type.

CC:0031.05.03.12.001 If the specified sensor type is not supported, a receiving node SHOULD set the *Scale Bit Mask* field to 0 in the returned response.

4.68.5 Multilevel Sensor Supported Scale Report Command

This command is used to advertise the supported scales of a specified multilevel sensor type.

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 bits)

This field is used to indicate the actual Sensor Type for which the supported scales are being advertised.

Scale Bit Mask (4 bits)

This field is used to advertise the supported scales for the actual sensor type.

CC:0031.05.06.11.001 This field MUST be treated as a bit mask and interpreted as follow:

- Bit 0 indicates support for the first scale of the actual Sensor Type
- Bit 1 indicates support for the second scale of the actual Sensor Type
- …

The list of scales for a given sensor type is defined in [17].

CC:0031.05.06.11.002 The value 1 MUST indicate that the corresponding scale is supported for the sensor type.
The value 0 MUST indicate that the corresponding scale is not supported for the sensor type.

Reserved

CC:0031.05.06.11.003 This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

4.68.6 Multilevel Sensor Get Command

This command is used to request the current reading from a multilevel sensor.

CC:0031.05.04.11.001 The Multilevel Sensor Report Command MUST be returned in response to this command.

CC:0031.05.04.11.002 This command MUST NOT be issued via multicast addressing.

CC:0031.05.04.11.003 A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods

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 bits)

This field is used to request a node to report a reading for the specified sensor type.

CC:0031.05.04.11.004 If this field is unspecified or a receiving node does not support the specified Sensor Type, it MUST reply with a pre-defined default Sensor Type and Scale.

Scale (2 bits)

This field is used to request a node to report a reading with a particular scale for the actual Sensor Type.

CC:0031.05.04.11.005 A node receiving a non-supported scale for the actual Sensor Type MUST reply with a supported scale within the Sensor type.

CC:0031.05.04.11.006 A sending node MUST ensure that the receiver supports the requested Sensor Types and/or Scales using the Multilevel Sensor Get Supported Sensor/Scale commands.

Reserved

CC:0031.05.04.11.007 These fields MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

4.68.7 Multilevel Sensor Report Command

This command is used to advertise a multilevel sensor reading.

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							

Sensor Type (8 bits)

This field is used to specify the sensor type of the actual sensor reading.

CC:0031.05.05.11.001 This field MUST be set to a value defined in [17]. Values not defined in [17] are reserved and MUST NOT be used.

CC:0031.05.05.11.002 A node MUST support as a minimum the highest Multilevel Sensor Command Class version associated with the Scale and Type it supports. The minimum required version for each Scale and Type is specified in [17].

Precision (3 bits)

CC:0031.05.05.11.003 This field is used to indicate how many decimal places are included the Sensor Value field. For example, the Sensor Value 1025 with precision 2 MUST be interpreted as equal to 10.25.

Scale (2 bits)

This field is used to indicate what scale is used for the actual sensor reading.

CC:0031.05.05.11.004 This field MUST be set to a value defined in [17]. Values not defined in [17] are reserved and MUST NOT be used.

CC:0031.05.05.11.005 A node MUST implement as a minimum the highest Multilevel Sensor Command Class version associated with the Scale and Type it supports. The minimum required version for each Scale and Type is specified in [17].

Size (3 bits)

This field is used to indicate the length in bytes of the Sensor Value field.

CC:0031.05.05.11.006 This field MUST be set to 1, 2 or 4.

Sensor Value (N bytes)

This field is used to advertise the value of the actual sensor reading.

CC:0031.05.05.11.007 The length of this field MUST be according to the Size field value.

CC:0031.05.05.11.008 The first byte MUST be the most significant byte.

CC:0031.05.05.11.009 This field MUST be encoded using signed representation and comply with Table 10, Signed field encoding (two's complement representation).

CC:0031.05.05.11.00A A controlling node receiving this command MUST always show the sensor value as is even though the Sensor Type and/or Scale are unknown.

CC:0031.05.05.12.001 A controlling node SHOULD implement the capability to update its list of Sensor Type and Scales, so that new Sensor Types and Scales added in [17] are not presented as unknown. If a controlling node receives an unknown Sensor Type or Scale, it SHOULD allow the user to assign a free-text description to the sensor reading.

CC:0031.05.05.12.002

4.68.7.1 Detailed description: Sensor Types for Movement and Rotation

A device may report position, velocity (position change over time) or acceleration (velocity change over time). Position, velocity and acceleration may all refer to a linear scale following an axis or a polar scale circling an axis. Position may be reported in an absolute or relative fashion.

The 3D reference coordinate system outlined below MUST be used for reporting changes in the physical orientation.

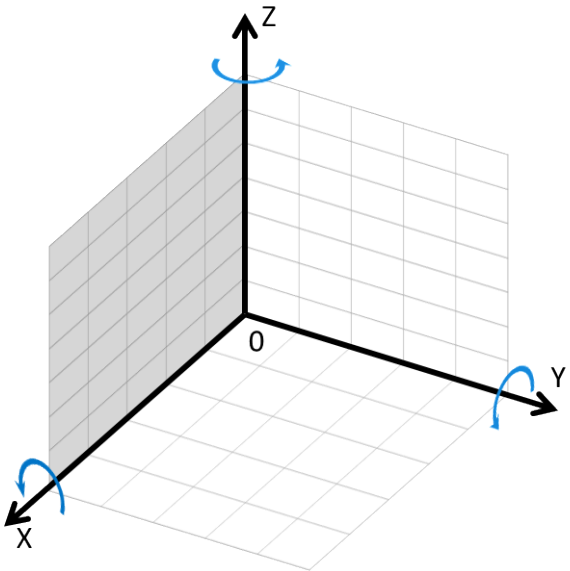


Figure 13, 3D reference coordinate system

CC:0031.05.05.11.00B Information relating to linear position, velocity and acceleration MUST refer to the zero position on a given axis. Thus a position change or a velocity MUST be positive if moving towards a larger position, measured from the zero position.

CC:0031.05.05.11.00C Information relating to an angle or change in angle MUST use the right-hand rule. This means that if one (virtually) grabs around an axis with the right hand, with the thumb in the direction of the axis, the angle increases in the direction of the index finger.

Table 78, Definition of position, velocity or acceleration

Application	Definition
Linear position, absolute	Position with reference to 0 (e.g. 1 meter)
Linear position, relative	Position with reference to previous position (e.g. 1 meter)
Linear velocity	Position change per time unit (e.g. 1 meter/second)
Linear acceleration	Velocity change per time unit (e.g. 1 meter/second^2)
Polar position, absolute	Angle with reference to 0 (e.g. 45 degrees)
Polar position, relative	Angle with reference to previous angle (e.g. 45 degrees)
Polar velocity (rotation)	Angle change per time unit (e.g. 1 degree/second)
Polar acceleration	Velocity change per time unit (e.g. 1 degree/second^2)

Depending on the number of axes supported by a given device, changes in the physical orientation are mapped to one, two or three axes as outlined in Table 79.

Table 79, Mapping of 1D, 2D and 3D movement and rotation

Dimensions	Movement	Rotation
1	Along X axis	Around X axis
2	Along X and Y axes	Around X and Y axes
3	Along X, Y and Z axes	Around X, Y and Z axes

A number of Sensor Types allow a device to report changes in the physical orientation.

Table 80, Recommended Sensor Types for reporting movement and rotation

Application	Sensor Type	Intended usage
1D Linear position, absolute	Distance (v3)	Single axis measurement of position (m)
1D Linear position, relative	*(no support)*	Single axis measurement of position change (m)
1D Linear velocity	Velocity (v2)	Single axis measurement of velocity (m/s)
1D Linear accelleration	Acceleration, X (v8)	Single axis measurement of acceleration (m/s^2)
1D Polar position, absolute	Direction (v2)	Single axis measurement of angle (degrees)
1D Polar position, relative	*(no support)*	Single axis measurement of angle change
1D Polar velocity	Rotation (v5)	Single axis measurement of velocity (RPM)
1D Polar accelleration	*(no support)*	Single axis measurement of acceleration ($degree/s^2$)
2D Linear position, absolute	*(no support)*	Two axis measurement of position (m)
2D Linear position, relative	*(no support)*	Two axis measurement of position change (m)
2D Linear velocity	*(no support)*	Two axis measurement of velocity (m/s)
2D Linear accelleration	Acceleration, X (v8), Acceleration, Y (v8)	Two axis measurement of acceleration (m/s^2)
2D Polar position, absolute	*(no support)*	Two axis measurement of angle (degrees from North)
2D Polar position, relative	*(no support)*	Two axis measurement of angle change
2D Polar velocity	*(no support)*	Two axis measurement of velocity (RPM)
2D Polar accelleration	*(no support)*	Two axis measurement of acceleration ($degree/s^2$)

Application	Sensor Type	Intended usage
3D Linear position, absolute	*(no support)*	Three axis measurement of position (m)
3D Linear position, relative	*(no support)*	Three axis measurement of position change (m)
3D Linear velocity	*(no support)*	Three axis measurement of velocity (m/s)
3D Linear accelleration	Acceleration, X (v8), Acceleration, Y (v8), Acceleration, Z (v8)	Three axis measurement of acceleration (m/s^2)
3D Polar position, absolute	*(no support)*	Three axis measurement of angle (degrees from North)
3D Polar position, relative	*(no support)*	Three axis measurement of angle change
3D Polar velocity	*(no support)*	Three axis measurement of velocity (RPM)
3D Polar accelleration	*(no support)*	Three axis measurement of acceleration ($degree/s^2$)

4.68.7.1.1 Sensor Type = Acceleration

The sensor types “Acceleration, X”, “Acceleration, Y” and “Acceleration, Z” are used to advertise the acceleration of a device along the X, Y and Z axes, respectively.

CC:0031.05.05.11.00D A one-dimensional device MUST report acceleration using the “Acceleration, X” type.

CC:0031.05.05.11.00E A two-dimensional device MUST report acceleration using the “Acceleration, X” and “Acceleration, Y” types.

CC:0031.05.05.11.00F The Scale used MUST be m/s^2. An Acceleration value reported with the Scale m/s^2 may be converted by a receiving node to a “g-force” level by using the formula below:

$$1g = 9.81 m/s^2$$

(g represents the unit of Earth Gravity; NOT the weight unit “gram”)

4.68.7.2 Detailed description: Smoke Density

Figure 14 shows the principle of how a photoelectric smoke detector works. As shown the smoke sensor uses the smoke to reflect the light onto a photo cell. So when no smoke is present, the light will not be reflected onto the photo cell. When smoke is present, the light will be reflected onto the photo cell. Dependent on the smoke density, more light will be reflected onto the photo cell. So by measuring the received light strength on the photo cell, it is possible to estimate the smoke density.

It is not possible to determine an accurate unit for this, as it is estimated from the light strength on the photo cell. So the exact particle density of the smoke is not known. Also the light strength interval measured on the photo cell, may vary between sensors. So it is decided to report the smoke density in percent, where 0% is no smoke and 100% is the maximum received light strength on the photo cell.

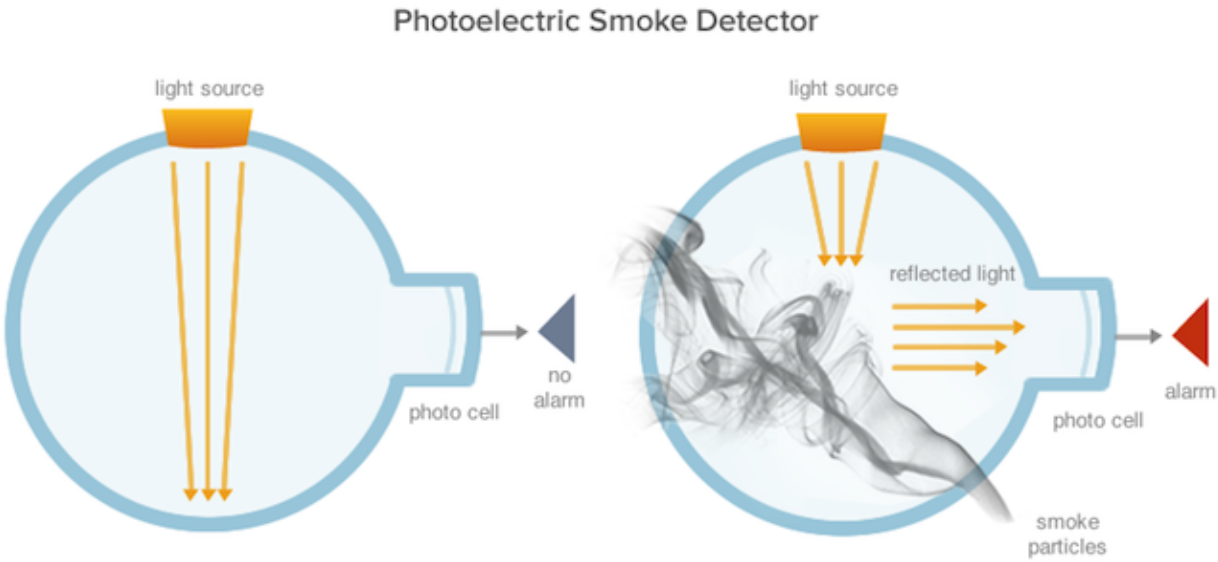


Figure 14, Photoelectric smoke detector

4.68.7.3 Detailed description: RF Signal Strength

The RF Signal Strength sensor type may report values using two different scales. While the dBm is a well-defined unit, the RSSI value represents a relative measurement where the internal sampling circuits and the actual sampling method is product specific.

CC:0031.05.05.11.010 The RSSI value MUST be reported in the range 0..100, where the value 100 represents the highest power level that can be measured.

4.69 Multilevel Switch Command Class, version 1

The Multilevel Switch Command Class is used to control devices with multilevel capability.

The Multilevel Switch Command Class is an actuator control command class. Refer to 3.6.

4.69.1 Multilevel Switch Set Command

This command is used to set a multilevel value in a supporting device.

The device MAY apply a non-zero duration to the transition from one value to a new value.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_MULTILEVEL							
Command = SWITCH_MULTILEVEL_SET							
Value							

Value (8 bits)

The encoding of the Value field MUST be according to Table 81.

Table 81, Multilevel Switch Set :: Value

Value		Level	State
0	(0x00)	0%	Off
1..99	(0x01..0x63)	Lowest non-zero level .. 100%	On
…		*Reserved*	*Reserved*
255	(0xFF)	Restore most recent (non-zero) level.	On

All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

The above mapping of the Multilevel Switch Command Class Value to hardware levels allows a controlling device to control a mixed group of Binary Switch and Multilevel Switch devices via Basic Set commands. Devices implementing the Binary Switch CC turn On or Off while devices implementing the Multilevel Switch CC sets the specified level.

The values 0x00 and 0xFF are special values which MUST be treated as state control commands indicating “Off” and “On”, respectively. A supporting device MUST restore the most recent (non-zero) value in response to the “On” state control command.

A device MAY implement up to 100 hardware levels (including 0). If a device implements less than 100 hardware levels, the hardware levels SHOULD be distributed uniformly over the entire range.
The mapping of command values to hardware levels MUST be monotonous, i.e. a higher value MUST be mapped to either the same or a higher hardware level. Refer to 3.6.2.

4.69.2 Multilevel Switch Get Command

This command is used to request the status of a multilevel device.

The Multilevel Switch Report command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_MULTILEVEL							
Command = SWITCH_MULTILEVEL_GET							

4.69.3 Multilevel Switch Report Command

This commandis used to advertise the status of a multilevel device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_MULTILEVEL							
Command = SWITCH_MULTILEVEL_REPORT							
Value							

Value (8 bits)

The encoding of the Value field MUST be according to Table 82.

Table 82, Multilevel Switch Report :: Value

Value		Hardware level	State
0	(0x00)	0%	Off
1..99	(0x01..0x63)	Lowest non-zero level .. 100%	On
...		*Reserved*	On
254	(0xFE)	Unknown	Unknown
255	(0xFF)	(100%)	On [Deprecated]

All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

The value 255 (0xFF) has been deprecated as it provides no information on the actual value in the device. A sending node MUST NOT use the value 255. A receiving node MUST interpret the value 255 as 100%.

The Value field SHOULD advertise the current value of the device hardware; also while in transition to a new target value.

A controlling device MUST NOT assume that the Value is identical to a value previously issued with a Set command when a transition has ended.

4.69.4 Multilevel Switch Start Level Change Command

This command is used to initiate a transition to a new level.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_MULTILEVEL							
Command = SWITCH_MULTILEVEL_START_LEVEL_CHANGE							
Reserved	Up/ Down	Ignore Start Level	Reserved				
Start Level							

Up/Down (1 bit)

This field MUST specify the direction of the level change.

If the Up/Down bit is set to 0 the level change MUST be increasing.
If the Up/Down bit is set to 1 the level change MUST be decreasing.

Ignore Start Level (1 bit)

A receiving device SHOULD respect the start level if the Ignore Start Level bit is 0.
A receiving device MUST ignore the start level if the Ignore Start Level bit is 1.

A controlling device SHOULD set the Ignore Start Level bit to 1.

Reserved

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Start Level (8 bits)

The Start Level field MUST specify the initial level of the level change.

4.69.5 Multilevel Switch Stop Level Change Command

This command is used to stop an ongoing transition.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_MULTILEVEL							
Command = SWITCH_MULTILEVEL_STOP_LEVEL_CHANGE							

4.70 Multilevel Switch Command Class, version 2

The Multilevel Switch Command Class is used to control devices with multilevel capability.

The Multilevel Switch Command Class is an actuator control command class. Refer to 3.6.

4.70.1 Compatibility considerations

A device supporting Multilevel Switch CC, version 2 MUST support Multilevel Switch CC, version 1.

Version 2 adds a “Duration” parameter to the Multilevel Switch Set and Multilevel Switch Start/Stop Level Change commands.

Commands not described in Version 2 remain unchanged from Version 1.

4.70.2 Multilevel Switch Set Command

This command is used to set a multilevel value in a supporting device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_MULTILEVEL							
Command = SWITCH_MULTILEVEL_SET							
Value							
Duration							

Value (8 bits)

Refer to Multilevel Switch Set V1 command.

Duration (8 bits)

The Duration field MUST specify the time that the transition should take from the current value to the new target value.
A supporting device SHOULD respect the specified Duration value.

The encoding of the Duration field MUST be according to Table 7.

The factory default duration SHOULD be the same as the duration used for the Multilevel Switch Set command, version 1.

4.70.3 Multilevel Switch Start Level Change Command

This command is used to initiate a transition to a new level.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_MULTILEVEL							
Command = SWITCH_MULTILEVEL_START_LEVEL_CHANGE							
Reserved	Up/ Down	Ignore Start Level	Reserved				
Start Level							
Duration							

Up/Down (1 bit)

This field MUST specify the direction of the level change.

If the Up/Down bit is set to 0 the level change MUST be increasing.
If the Up/Down bit is set to 1 the level change MUST be decreasing.

Ignore Start Level (1 bit)

A receiving device SHOULD respect the start level if the Ignore Start Level bit is 0.
A receiving device MUST ignore the start level if the Ignore Start Level bit is 1.

A controlling device SHOULD set the Ignore Start Level bit to 1.

Reserved

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Start Level (8 bits)

The Start Level field MUST specify the initial level of the level change.

Duration (8 bits)

The dimming rate to use MUST be calculated to match a transition from 0 to 99 during the time specified by the Duration field.
A supporting device SHOULD respect the specified Duration value.

For encoding of the duration value refer to section 4.70.2.

4.71 Multilevel Switch Command Class, version 3

The Secondary Switch Type of the Multilevel Switch Command Class, version 3 has been DEPRECATED.

The Primary Switch Type 0x00, indicating "Not supported", of the Multilevel Switch Command Class, version 3 has been OBSOLETED.

The implementation of Secondary Switch Type functionality is NOT RECOMMENDED.

While the functionality related to the Secondary Switch Type of the Multilevel Switch Command Class, version 3 is deprecated, a supporting device claiming compliance with this version MUST implement support for the Multilevel Switch Supported Get Command.

For backwards compatibility reasons, a device MAY implement Secondary Switch Type functionality. It is however RECOMMENDED that Multi Channel Command Class support is also implemented if a device provides multiple controllable resources in the same physical entity.

The Multilevel Switch Command Class is used to control devices with multilevel capability.

The Multilevel Switch Command Class is an actuator control Command Class. Refer to 3.6.

Multilevel Switch CC, version 3 adds two-dimensional resource control capabilities to the Multilevel Switch Start/Stop Level Change commands and introduces two new commands for the discovery of those capabilities.

The Secondary Switch Type is intended for devices providing two-dimensional control, e.g. Window blinds, where the Primary Switch Type is used for Up/Down control and the Secondary Switch Type is used for controlling the slat tilt angle.

The Multilevel Switch Set, Get and Report commands MUST address the primary device functionality. The Multilevel Switch Start/Stop Level Change commands MUST manipulate the Primary Switch Type functionality based on the Up/Down parameter.

The Multilevel Switch Start/Stop Level Change commands MUST manipulate the Secondary Switch Type functionality based on the Inc/Dec parameter.
If the Secondary Switch Type is 0x00 (Undefined / Not supported), the Multilevel Switch Start/Stop Level Change (Inc/Dec) command parameter MUST be ignored.

4.71.1 Compatibility considerations

A device supporting Multilevel Switch CC, version 3 MUST support Multilevel Switch CC, version 2.

A device supporting Multilevel Switch CC, version 3 MUST implement the Primary Switch type.

Commands not described in Version 3 remain unchanged from Version 2.

4.71.2 Multilevel Switch Supported Get Command

This command is used to request the supported Switch Types of a supporting device.

The Multilevel Switch Supported Report command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_MULTILEVEL							
Command = SWITCH_MULTILEVEL_SUPPORTED_GET							

4.71.3 Multilevel Switch Supported Report Command

This command is used to advertise the supported Switch Types implemented by a supporting device.

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

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Primary Switch Type (5 bits)

The Primary Switch Type field MUST represent the primary device functionality.

Primary Switch Type 0x00, indicating “Not supported”, has been OBSOLETED.

Previous revisions of this specification allowed that a device may (theoretically) be implemented with no primary device functionality, i.e. Primary Switch Type 0x00, and specified that such a device should indicate this situation by returning Multilevel Switch Report commands carrying the value 0xFE. The value 0xFE is incompatible with Versions 1 and 2 of the Multilevel Switch Command Class and MUST NOT be used in a Multilevel Switch Report.

The Primary Switch Type MUST comply with Table 83.

A supporting device MUST implement the Primary Switch type.
The Primary Switch Type SHOULD be 0x02 (Up/Down).
The Primary Switch Type MUST NOT be 0x00 (Undefined).

Table 83, Encoding of Primary and Secondary Switch Types

Switch Type Value	0x00 (Direction/Endpoint A)	0x63/0xFF (Direction/Endpoint B)
0x00	Undefined / Not supported (Secondary only)	
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	

All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

Secondary Switch Type (5 bits)

The Secondary Switch Type field MUST represent the secondary device functionality. The Secondary Switch Type MUST comply with Table 83

A supporting device MAY implement the Secondary Switch type.

4.71.4 Multilevel Switch Start Level Change Command

This command is used to initiate a transition to a new level.

The Multilevel Switch Command Class, version 3 adds a “Secondary Switch Inc/Dec” and a “Secondary Switch Step Size” field to this command to support motor controlled devices featuring two-dimensional motion.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_MULTILEVEL							
Command = SWITCH_MULTILEVEL_START_LEVEL_CHANGE							
Primary Switch Up/Down		Ignore Start Level		Secondary Switch Inc/Dec		Reserved	
Primary Switch Start Level							
Duration							
Secondary Switch Step Size							

Primary Switch Up/Down (2 bits)

The Up/Down field MUST be used for manipulating the primary device functionality.

This field MUST be encoded according to Table 84.

Table 84, Encoding of the Up/Down field

Value	Description	Details
0x00	Up	Increase level for Primary Switch Type
0x01	Down	Decrease level for Primary Switch Type
0x02	*Reserved*	
0x03	No Up/Down motion	Maintain current level for Primary Switch Type

Ignore Start Level (1 bit)

A receiving device SHOULD respect the start level if the Ignore Start Level bit is 0.
A receiving device MUST ignore the start level if the Ignore Start Level bit is 1.

Secondary Switch Inc/Dec (2 bits)

The Inc/Dec field MUST be used for controlling the secondary device functionality.

If the Secondary Switch Type is 0x00 (Undefined / Not supported), the Inc/Dec field MUST be ignored.

This field MUST be encoded according to Table 85.

Table 85, Encoding of the Inc/Dec field

Value	Description	Details
0x00	Increment	Increase level for Secondary Switch Type
0x01	Decrement	Decrease level for Secondary Switch Type
0x02	*Reserved*	
0x03	No Inc/Dec	Maintain current level for Secondary Switch Type

As this field defines the value “00” of two previously reserved bits as the code “Increment”, a supporting device implementing Multilevel Switch CC, version 3 MUST correctly identify the Multilevel Switch Start Level Change command to be version 3 (by the presence of the <Secondary Switch Step Size> field) before interpreting the <Secondary Switch Inc/Dec> field. Failing to do so will cause a device to start incrementing the Secondary Switch Type in response to version 1 and version 2 variants of the Multilevel Switch Start Level Change command.

Reserved

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Primary Switch Start Level (8 bits)

The Start Level field MUST be used for controlling the primary device functionality.

Duration (8 bits)

This field is unchanged from version 2.

Secondary Switch Step Size (8 bits)

The Step Size field MAY be used for controlling a secondary device functionality.
If the Secondary Switch Type is 0x00 (Undefined / Not supported), the Step Size field MUST be ignored.
If the Secondary Switch Type is not 0x00, the Step Size field MUST be used for controlling the secondary device functionality.

If the Inc/Dec field is set to 3 (No Inc/Dec), the Step Size field MUST be set to 0.

This field MUST carry a value in the range {0x00..0x63, 0xFF}. All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.
A receiving device MUST accept any of the values in the above range. 0x01 MUST represent the lowest non-zero level and 0x63 MUST represent the highest level

A device MAY implement 100 hardware levels (including 0). If a device implements less than 100 hardware levels, the mapping to hardware levels SHOULD be distributed equally over the entire range of 1..99 (0x01..0x63).
The mapping of command values to hardware levels MUST be monotonous, i.e. a higher value MUST be mapped to either the same or a higher hardware level.

An implementation MUST interpret the combined Inc/Dec and Step Size fields as outlined in Table 86.

Table 86, Interpretation of the Inc/Dec and Step Size fields

(Secondary Switch Type)	Inc/Dec	Step Size	Interpretation
(undefined)	*(ignore)*	*(ignore)*	Secondary Switch Type is undefined ⇒ ignore fields
0x01..0x07	No Inc/Dec	*(ignore)*	No Inc/Dec ⇒ Maintain current Secondary Switch level
0x01..0x07	Increment	x	Increase Secondary Switch level by x steps
0x01..0x07	Decrement	y	Decrease Secondary Switch level by y steps

The requested level change SHOULD take the time specified by the Duration field.

4.72 Multilevel Switch Command Class, version 4

The Multilevel Switch Command Class is used to control devices with multilevel capability.

The Multilevel Switch Command Class is an actuator control Command Class. Refer to 3.6.

4.72.1 Compatibility considerations

Version 4 adds reporting of target value and duration.

A device supporting Multilevel Switch CC, Version 4 MUST support Multilevel Switch CC, Version 3.

A device receiving a V1 Multilevel Set command MAY apply a factory default duration to the transition to a new value.

4.72.2 Multilevel Switch Report Command

This command is used to advertise the status of a multilevel device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_MULTILEVEL							
Command = SWITCH_MULTILEVEL_REPORT							
Current Value							
Target Value							
Duration							

Current Value (8 bits)

The encoding of the Value field MUST be according to Table 82.

The device may be queried for its current value while in transition to a new value. The response to a Get command SHOULD be the current value of the device hardware.

A controlling device MUST NOT assume that the Value is identical to a value previously issued with a Set command when a transition has ended.

Target Value (8 bits)

The Target Value field MUST advertise the target value of an ongoing transition or the most recent transition.

The encoding of the Target Value field MUST be according to Table 82.

If queried after receiving a Set command, the Target Value field MUST advertise the target value specified in the Set command. The Target Value may change at a later time due to local control or a Multilevel Switch Stop Level Change command.

If the device is in a motion controlled transition, the Target Value field MUST advertise the value 0x00 or 0x63.

Duration (8 bits)

The Duration field SHOULD advertise the time needed to reach the Target Value at the actual transition rate. The encoding of the Duration field MUST be according to Table 8.

4.73 Multilevel Toggle Switch Command Class, version 1 [DEPRECATED]

THIS COMMAND CLASS HAS BEEN DEPRECATED

A device MAY implement this command class, but it is RECOMMENDED that new implementations use the Multilevel Switch Command Class.
If implementing this command class, it is RECOMMENDED that the Multilevel Switch Command Class is also implemented.

The Multilevel Toggle Switch Command Class is used for multilevel toggle-style actuator devices.

4.73.1 Multilevel Toggle Switch Set Command

This command is 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							

4.73.2 Multilevel Toggle Switch Get Command

This command is used to request the state of the load controlled by the device.

The Multilevel Toggle Switch Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_TOGGLE_MULTILEVEL							
Command = SWITCH_TOGGLE_MULTILEVEL_GET							

4.73.3 Multilevel Toggle Switch Report Command

This command is used to advertise the level of a toggle switch.

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 MAY be 0x00 (off/disable) or 0xFF (on/enable).
The field MAY carry values from 1 to 99.

4.73.4 Multilevel Toggle Switch Start Level Change Command

This command is 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, the switch SHOULD stop when reaching the max or min level. If the roll over bit is set to 1, 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

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Start Level (8 bits)

The Start Level field contains the initial level that the switch should assume when it start to change the level.

4.73.5 Multilevel Toggle Switch Stop Level Change Command

This command is 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							

4.74 Notification Command Class, version 3-8

THIS COMMAND CLASS VERSIONS 3-7 HAVE BEEN DEPRECATED

CC:0071.03.00.13.001
CC:0071.03.00.12.001

A device MAY implement version 3 to 7, but it is RECOMMENDED that new implementations comply with Notification Command Class, version 8

CC:0071.03.00.12.002

Pull Mode has been deprecated; it is RECOMMENDED that new implementations support Push Mode

The Notification Command Class is used to advertise events or states, such as movement detection, door open/close or system failure. The Notification Command Class supersedes the Alarm Command Class.

4.74.1 Terminology

Sensors may be designed for several purposes. A multilevel sensor advertises measurements or readings. A notification sensor sends event or state notifications.

This Command Class is used for notification sensors. The Multilevel Sensor Command Class is used for multilevel sensors.

Notifications are categorized into logical groups called Notification Types. A Notification is denoted with its type and event/state: {Notification Type::event/state}. A node may send Notifications from several Notification Types.

An event only has a meaning in the moment it happens. An event does not indicate the value of a state variable. An example is given in Figure 15

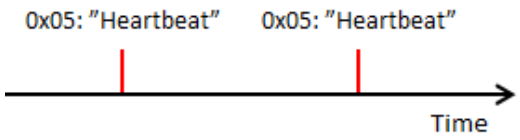


Figure 15, Event notifications inform only about instantaneous situations

A state variable may assume two or more states. Some state variables are returned to their idle state via a generic "State idle" Notification.

For example, a controlling node receiving a {Smoke alarm::Smoke detected} Notification will consider that the state has not changed until it receives a {Smoke alarm::State idle(Smoke detected)} Notification. An illustration is given in Figure 16.

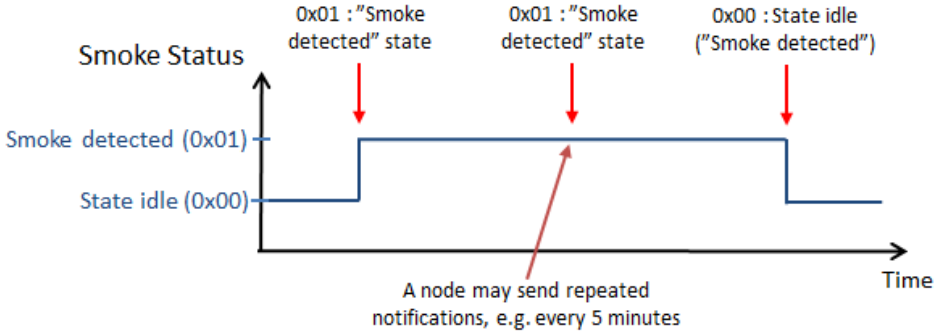


Figure 16, Binary state variable with generic "State idle" Notification

The "State idle" notification may also be used to return a multi-value state variable to its idle state. An example is given inFigure 17.

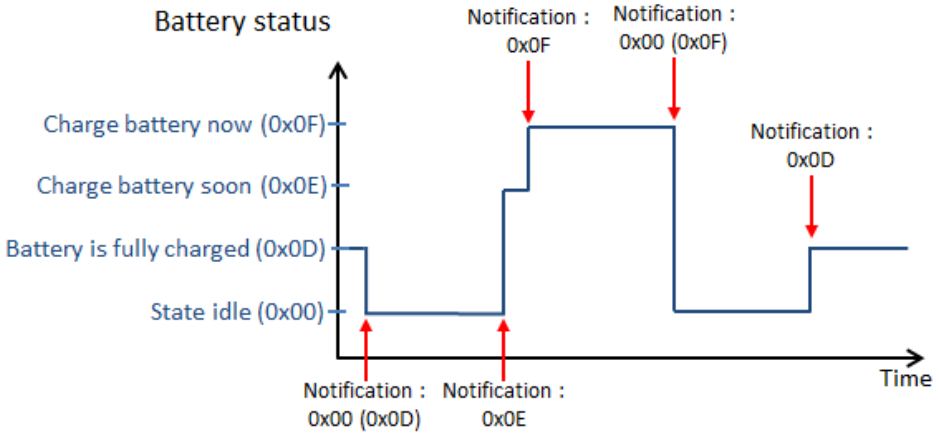


Figure 17, Multi-value state variable with generic "State idle" Notification

Some state variables use a specific Notification defined for each state change. Figure 18 shows an example of such a state variable.

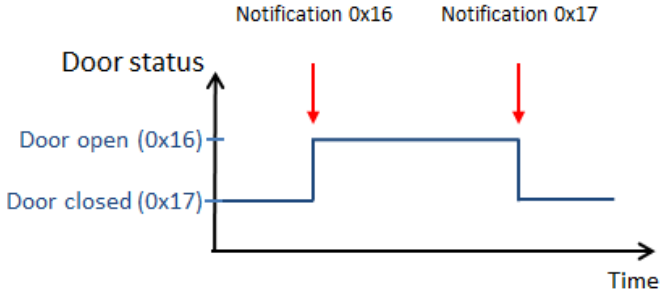


Figure 18, Binary state variable with specific idle state Notification

Some notifications are complemented with event/state parameters. For example, the {System::System software failure} Notification may be accompanied with the manufacturer's failure/error codes as event parameters.

Event/state parameters may also affect state variables and their states. For example, the Notification {Irrigation::Schedule started} takes a 1-byte parameter which identifies the Schedule ID that is started. In that case, each state variable is identified using the state parameters, thus creating a state variable array.

An illustration is given in Figure 19, where the same Notifications with different parameters advertise the state of different state variables.

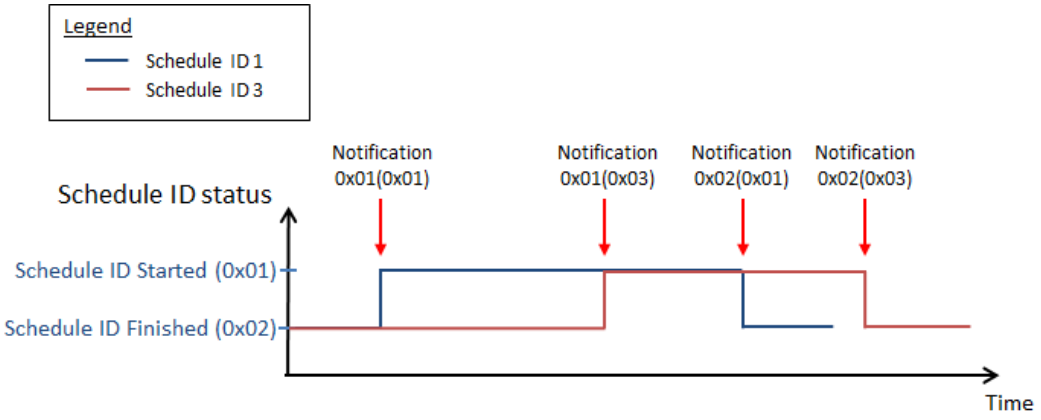


Figure 19, State variable array defined by additional Event/State Parameters

A complete list of Notifications (including state variables, event parameters, etc.) is given in [18].

A notification node operates either in Push or in Pull mode.

A Push node sends unsolicited Notifications. The transmission of unsolicited notifications may be disabled or enabled. When enabled, unsolicited Notifications are transmitted via an Association Group. It is not possible to subsequently retrieve Notifications issued by a Push node.

A Pull sensor collects events and state changes in a queue of pending Notifications. Notifications are retrieved one by one from the Pull sensor queue via the Notification Get Command. The Pull sensor advertises that its queue is empty when all Notifications have been retrieved.
A persistent Pull Notification is not removed from the Pull sensor queue until it is actively cleared by a controlling node.

4.74.2 Compatibility considerations

4.74.2.1 Notifications and Command Class version

CC:0071.03.00.21.001 New Notification Types and Notifications have been added to each new version of this Command Class. A node MUST implement as a minimum the Notification Command Class version associated with the Notifications it sends. The minimum required version for each Notification is specified in [18].

4.74.2.2 Push mode requirements

CC:0071.03.00.22.001 A node supporting the Notification Command Class SHOULD implement Push mode.

CC:0071.03.00.21.002 A Push node MUST implement the Association Command Class.

CC:0071.03.00.22.002 A Push node SHOULD advertise Association Groups through the Association Group Information (AGI) Command Class.

CC:0071.03.00.21.003 A Z-Wave Plus Push node MUST:

- Provide all supported Notifications via the Lifeline Association group.
- Advertise Association Groups through the Association Group Information (AGI) Command Class.

4.74.2.3 Pull mode requirements

Earlier text revisions presented inconsistencies and undefined behaviors for Pull nodes. Thus, existing Pull nodes may behave differently than expected by a controlling node. A node supporting the Notification Command Class SHOULD NOT implement Pull mode.
CC:0071.03.00.22.003

CC:0071.03.00.23.001
CC:0071.03.00.22.004 A Pull node MAY reorder Notifications according to priority so that the first detected event is not the first to be reported. A Pull node SHOULD NOT reorder states of the same state variable in the event queue. For example, the “Program in progress” and “Program completed” Notifications SHOULD stay in the same order.

CC:0071.03.00.22.005 A Pull node having its queue full SHOULD remove the oldest notification entry from the queue.

CC:0071.03.00.22.006 Persistent notifications SHOULD carry a sequence number.

4.74.2.4 Multi Channel considerations

CC:0071.03.00.21.004 If several End Points within a Multi Channel device support the Notification Command Class, they MUST all operate in the same mode (i.e. either all End Points operate in Push mode or all End Points operate in Pull mode).

4.74.2.4.1 Multi Channel Push nodes

CC:0071.03.00.23.002
CC:0071.03.00.21.005 While End Points MAY send identical notifications via the Root Device Lifeline Group, the Root Device MUST NOT send identical Notifications on behalf of multiple End Points. Illustrations are given in Figure 20 and Figure 21.

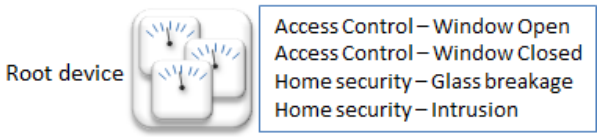


Figure 20, Multi Channel device aggregating Notifications (Lifeline group)

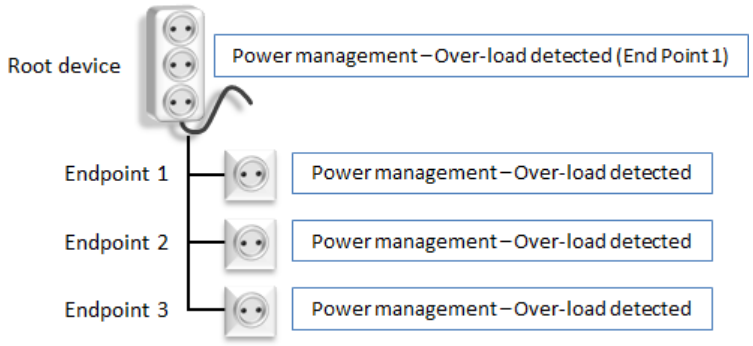


Figure 21, Multi Channel device with End Point notification overlap (Lifeline group)

CC:0071.03.00.23.003 A Multi Channel node with identical End Points issuing the same binary state notification with generic state idle Notification MAY be an exception and issue binary Notifications representing the state all End Points.

CC:0071.03.00.21.008 If doing so, the Root Device MUST issue the active state notification if any of the End Points has the active binary state and MUST issue the event inactive Notification when all End Points are back to the "State idle".

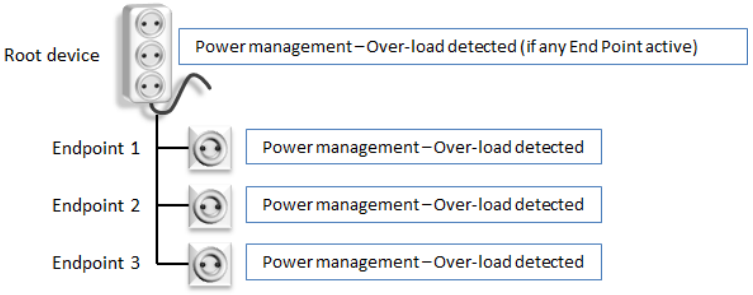


Figure 22, Notification representing all End Points for Binary State with generic State Idle notification

CC:0071.03.00.21.006 Z-Wave Plus Multi Channel devices MUST send End Point Push notifications via the Root Device Lifeline Association Group (if notifications are not identical).

CC:0071.03.00.22.007 A controlling node SHOULD create a Multi Channel Association to the Lifeline group of a supporting node. Thus Notifications will be sent Multi Channel encapsulated via the Root Device Lifeline group, allowing End Points to send identical Notifications.

Individual End Point Notifications can be enabled/disabled by sending a Multi Channel encapsulated Notification Set Command to an End Point, even when End Point Notifications are actually sent via the Root Device Lifeline group.

4.74.2.5 State idle

Every Notification Type has the same generic Notification 0x00 "State idle", (also known as "Event Inactive" in older specification text) which was introduced in version 4. The Notification allows the device to advertise that a state variable returned to its idle state.

CC:0071.04.00.23.001 A Push sensor MAY send repeated state Notifications without sending any "State idle" Notification to indicate that a given state is still active. For instance, a smoke sensor can send a "Smoke Detected" Notification every five minutes to indicate that the state variable has not returned to idle (refer to Figure 16.).

CC:0071.08.00.21.001 Notification Command Class, version 8 increases the requirement level for the use of the “State idle” Notification from OPTIONAL to REQUIRED. Refer to 4.74.2.11.

CC:0071.04.00.22.001 A supporting node SHOULD NOT send a “State idle” notification for an event or for a state variable to which state idle does not apply. Refer to [18] for state variables to which the “State idle” notification applies

4.74.2.6 Version 3 [DEPRECATED]

The Notification Command Class version 3 is an extension of the Alarm Command Class version 2 and adds the following:

- Additional Notifications (refer to [18])
- Interview process includes Events (Event Supported Get / Event Supported Report)
- Sequence field added for collection management of reports
- The queue empty status (=0xFE) for Pull nodes. A Pull node can now advertise Notification Status = “no pending notifications”

Commands not mentioned in this version are unchanged from Alarm Command Class version 1 and/or Alarm Command Class version 2.

The CC identifier for Notification CC V3 is the same as the Alarm CC V1 and V2. However, the Notification Command Class is not fully backwards compatible with the Alarm Command Class, versions 1 and 2. Clarifications for ensuring backwards compatibility with version 1 and 2 are given in version 4 of this Command Class.

CC:0071.03.00.21.007 An implementation supporting Alarm CC V1 fields MUST map proprietary alarm types and levels to a similar Notification Type and Notification CC V3 where possible. In addition, all Alarm CC V1 alarm types and levels MUST be described in the product manual.

4.74.2.7 Version 4 [DEPRECATED]

Version 4 of this Command Class introduces the following:

- Additional Notifications (refer to [18])
- Event/State = 0x00 (State idle) for a given Notification Type to indicate that all state variables of that Notification Type returned to idle
- The Zensor Net Source Node ID field has been discontinued. It is now a reserved field.
- Clarification to expected behavior in regards to:
  - V1 and V2 Alarm Get Command handling
  - Notification Status field description
  - Notification Type = 0xFF, “Return first detected notification on supported list”
  - Event = 0xFE in Event Supported Report Command, which must not be advertised.

The support for state idle (event/state=0x00) cannot be advertised in the Event Supported Get Command and version 4 nodes will appear to use reserved values for version 3 controlling nodes.

CC:0071.04.00.21.001 A version 4 node MUST comply with the rules outlined in Table 87 and Table 88.

Table 87 outlines the required behavior when receiving an Alarm Get command, Version 1.
Table 88 outlines the required behavior when receiving an Alarm Get command, Version 2.
The version of an Alarm Get Command can be determined from the command length.

Table 87, Required behavior when receiving Alarm Get, version 1

Received Command: V1, ALARM_GET (Alarm Type = x)	
V1 Alarm Type supported (x)?	**Response**
YES	V1, ALARM_REPORT (Alarm Type = x, Alarm Level = level)
NO	V1, ALARM_REPORT (Alarm Type = 0x00, Alarm Level = 0x00)

Table 88, Required behavior when receiving Alarm Get, version 2

Received Command: V2, ALARM_GET (Alarm Type = x, Z-Wave Alarm Type = y)			
	Z-Wave Alarm Type supported (y)?	**V1 Alarm Type supported (x)?**	**Response**
	NO	NO	NO RESPONSE
	NO	YES	V2, ALARM_REPORT (Alarm Type = x Alarm Level = level Reserved = 0x00 Z-Wave Alarm Status = 0x00 Z-Wave Alarm Type = 0x00 Z-Wave Alarm Event = 0x00 Number of Event Param = 0x00)
CC:0071.04.00.21.002	YES or y = 0xFF	YES/NO	Notifications newer than V2, idle states or empty queues MUST be represented with the unknown notification (Notification event/state field set to 0xFE).
CC:0071.04.00.21.003			If no state is detected (push nodes) or no event/state queued (pull nodes), the Notification Type MUST be set to one of the supported Notification Type in response to a Get (Type = 0xFF)
CC:0071.04.00.21.004			Push nodes: A supporting node MUST return the current states compatible with V2. Table 90 shows a Push node example returning responses to different V2 Alarm Get Commands.
CC:0071.04.00.21.005			
CC:0071.04.00.21.006			Pull nodes: A supporting node MUST return only V2 notifications from its queue when receiving a V2 Alarm Get Command. A supporting node MUST set the status to 0x00 and event to 0xFE when its queue is empty. Table 91 shows a Pull node example returning responses to different V2 Alarm Get Commands.

Table 90 and Table 91 show the responses of a node supporting Notification Types and Notifications described in Table 89:

Table 89, V2 Alarm Get node capabilities example

Node capabilities	
Notifications Types	**Notifications**
0x01: Smoke Alarm	0x01: Smoke detected 0x03: Smoke alarm test 0x06: Alarm silenced
0x03: CO2 Alarm	0x01: Carbon dioxide detected

Table 90, Push node responses to V2 Alarm Get (example)

Received Get		Current states at the receiving node			Returned response		
V1 Alarm Type	Type	Smoke Alarm:: Sensor status	Smoke Alarm:: Alarm status	CO2 Alarm:: Sensor status	Type	Status	Event/ State
-*	0x01	Idle	Idle	-	0x01	0x00 or 0xFF	0xFE
-	0x01	Smoke detected (V2)	Idle	-	0x01	0x00 or 0xFF	0x01
-	0x01	Smoke detected (V2)	Alarm silenced (V8)	-	0x01	0x00 or 0xFF	0x01
-	0x01	Idle	Smoke alarm test (V3)	-	0x01	0x00 or 0xFF	0xFE
0x00	0x02	-	-	-	No Report		
0x01	0x02	-	-	-	No Report, unless V1 alarm is supported. Refer to Table 88		
-	0xFF	Idle	Idle	Idle	0x01 or 0x03**	0x00 or 0xFF	0xFE
-	0xFF	Smoke detected (V2)	Idle	Idle	0x01	0x00 or 0xFF	0x01
-	0xFF	Smoke detected (V2)	Alarm silenced (V8)	Idle	0x01	0x00 or 0xFF	0x01
-	0xFF	Idle	Smoke alarm test (V3)	Idle	0x01	0x00 or 0xFF	0xFE
-	0xFF	Smoke detected (V2)	Idle	Carbon dioxide detected (V2)	0x01 or 0x03**	0x00 or 0xFF	0x01

*) The symbol '-' indicates that the value has no impact on the V2 fields of the returned response

**) It is up to the node to decide which Notification Type to report.

Table 91, Pull node responses to V2 Alarm Get (example)

Received Get		Before returning a report	Returned response			After returning a report
V1 Alarm Type	Type	Node's queue	Type	Status	Event/ State	Node's queue
-*	0x01	Empty	0x01	0x00	0xFE	Empty
-	0x01	Carbon dioxide detected (V2) Smoke detected (V2)	0x01	0x00	0x01	Carbon dioxide detected (V2)
-	0x01	Alarm silenced (V8) Smoke detected (V2)	0x01	0x00	0x01	Alarm silenced (V8)
-	0x01	Alarm silenced (V8)	0x01	0x00	0xFE	Alarm silenced (V8)
0x00	0x02	-	No Report			-
0x01	0x02	-	No Report, unless V1 alarm is supported. Refer to Table 88			-
-	0xFF	Empty	0x01 or 0x03**	0x00	0xFE	Empty
-	0xFF	Carbon dioxide detected (V2) Smoke detected (V2)	0x03	0x00	0x01	Smoke detected (V2)
-	0xFF	Alarm silenced (V8)	0x01	0x00	0xFE	Alarm silenced (V8)

*) The symbol '-' indicates that the value has no impact on the V2 fields of the returned response
**) It is up to the node to decide which Notification Type to report.

4.74.2.8 Version 5 [DEPRECATED]

Version 5 of this Command Class introduces additional Notifications (refer to [18]) and the event/state parameters for the special purpose State idle 0x00, allowing to specify which state variable returned to idle.

A version 4 controlling node will conclude that all state variables have returned to idle when a version 5 node advertises that a single state variable has returned to idle.

4.74.2.9 Version 6 [DEPRECATED]

Version 6 of this Command Class introduces additional Notifications (refer to [18])

Version 6 clarifies how to use the 'User Code Report' Event Parameter. The User Code Report Command is used as Event Parameter for the Notification {Access Control::Keypad Lock/Unlock Operation} Command.

An example of Notification Event/State Parameter encapsulation is given in 4.74.6.1

4.74.2.10 Version 7 [DEPRECATED]

Version 7 of this Command Class introduces additional Notifications (refer to [18]).

4.74.2.11 Version 8

CC:0071.08.00.22.001 It is RECOMMENDED that new nodes support version 8 of this Command Class.

CC:0071.08.00.23.001 Requirements for backwards compatibility with version 2 nodes introduced in version 4 have been found to be ambiguous and challenging to be observed correctly by newer nodes. Therefore, it is OPTIONAL for a version 8 node to comply with Table 88 when receiving a V2 Get Command.

Version 8 of this Command Class introduces additional Notifications (refer to [18]).

CC:0071.08.00.21.002 Version 8 increases the requirement level for the use of the “State idle” Notification from OPTIONAL to REQUIRED.

CC:0071.08.00.21.003 Supporting nodes implementing version 8 or newer of the Notification Command Class MUST issue a “State idle” Notification when a state variable returns to idle. For instance, the {Smoke alarm::Smoke

CC:0071.08.00.21.004 Detected} Notification MUST be followed by a {Smoke alarm::State idle(Smoke detected)} Notification when the smoke is no longer detected by the sensor. (refer to Figure 16).

4.74.3 Interoperability considerations

4.74.3.1 Push nodes Basic control

CC:0071.03.00.32.001 For Push nodes, it is RECOMMENDED to implement additional association groups for relevant Notifications, which issue a Basic Set Command. It allows configuring a device to control other nodes directly, e.g., turn on lights when motion is detected and turn off when there is no more motion.

4.74.3.2 Event flood

CC:0071.03.00.32.002 Certain sensors may trigger repeatedly within a short amount of time. It is RECOMMENDED to implement timers to arbitrate the transmission of notifications. Illustrations are given in Figure 23 and Figure 24.

inactive timeout : Timeout used for State idle notification
refresh timeout : Timeout used for repeated notifications

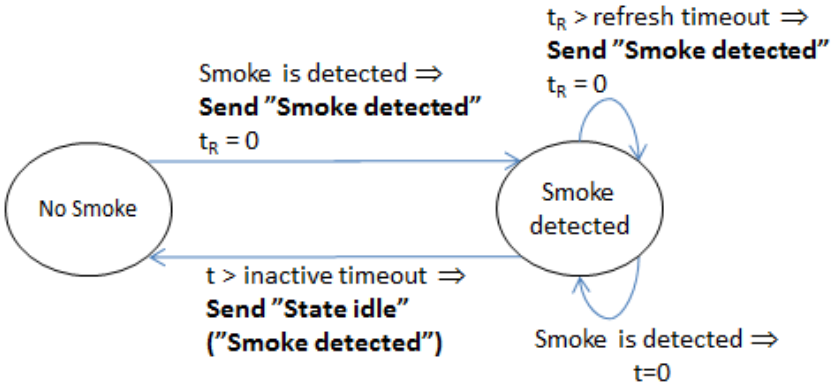


Figure 23, recommended timer mechanism for sending notifications (1)

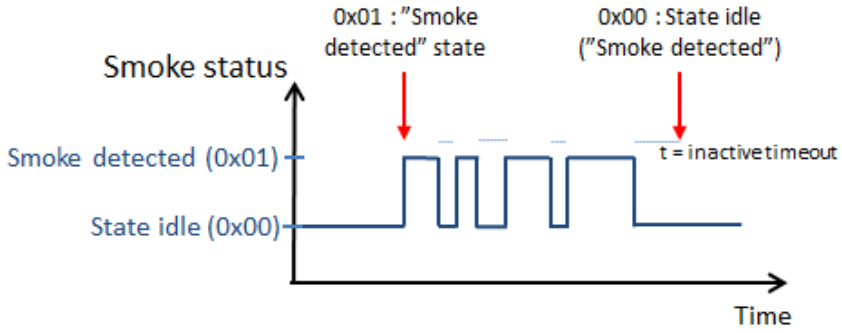


Figure 24, recommended timer mechanism for sending notifications (2)

4.74.4 Notification Set Command

Push nodes:

This command is used to enable or disable the unsolicited transmission of a specific Notification Type.

Pull nodes:

This command is used to clear a persistent Notification in the notification queue.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_NOTIFICATION							
Command = NOTIFICATION_SET							
Notification Type							
Notification Status							

Notification Type (8 bits)

This field is used to specify a Notification Type. Assigned values are defined in [18].

CC:0071.03.06.11.001 A sending node MUST specify a Notification Type that is supported by a receiving node.

CC:0071.03.06.11.002 A receiving node MUST ignore the command if this field is set to a non-supported Notification Type or if this field set to 0xFF.

Notification Status (8 bits)

Push nodes:

This field is used to set the Status of a Notification Type.

CC:0071.03.06.11.003 This field MUST comply with Table 92.

Table 92, Notification Set::Notification Status (push mode)

Value	Description	Version
CC:0071.03.06.11.004 0x00	Unsolicited messages MUST be disabled for the specified Notification Type	2
CC:0071.03.06.11.005 0xFF	Unsolicited messages MUST be enabled for the specified Notification Type	2

CC:0071.03.06.11.006 All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

CC:0071.03.06.13.001 A receiving node MAY deny the deactivation of a specific Notification Type. In that case, the
CC:0071.03.06.11.007 receiving node MUST respond to this command with an Application Rejected Request Command.
CC:0071.03.06.11.008 Thus, if a node can deny the deactivation of a Notification Type, the node MUST implement the
CC:0071.03.06.12.001 Application Status CC. A sending node should be aware that a receiving node may return a response to the Set Command and thus SHOULD apply a back-off timer before sending a subsequent command.

Pull nodes:

CC:0071.03.06.11.009 A sending node MUST set this field to 0x00 to indicate that a persistent Notification for the specified Notification Type MUST be cleared in the receiving node's queue.

CC:0071.03.06.11.00A All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

4.74.5 Notification Get Command

Push nodes:

This command is used to request if the unsolicited transmission of a specific Notification Type is enabled.

Some supporting nodes will also advertise a current state in response to this command.

Pull nodes:

This command is used to retrieve the next Notification from the receiving node's queue.

CC:0071.03.04.11.001 The Notification Report Command MUST be returned in response to this command unless this command is to be ignored. Refer to the fields description.

CC:0071.03.04.11.002 This command MUST NOT be issued via multicast addressing.

CC:0071.03.04.11.003 A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_NOTIFICATION							
Command = NOTIFICATION_GET							
V1 Alarm Type							
Notification Type							
Notification Event / State							

V1 Alarm Type (8 bits)

The use of this field depends on the V1 Alarm field advertised in the Alarm Type Supported Report Command.

A sending node MAY specify a V1 Alarm Type if the receiving node supports the actual alarm type.
A sending node MUST specify the value 0x00 if the receiving node does not support V1 alarms.

CC:0071.03.04.12.001 The receiving node behavior SHOULD comply with Table 87 and Table 88.

Notification Type (8 bits)

This field is used to specify a Notification Type. Assigned values are defined in [18].

CC:0071.03.04.11.004 This field MUST be set to a Notification Type that is supported by the receiving node or to 0xFF.

CC:0071.03.04.11.005 A receiving node MUST ignore the command if this field is different than 0xFF and set to a non-supported Notification Type.

Push nodes:

CC:0071.03.04.12.002 The value 0xFF indicates to the receiving node that it MUST select a supported Notification Type and SHOULD advertise the current state of one of its state variables within the chosen Notification Type.

CC:0071.03.04.11.006 A supported Notification Type value indicates that the receiving node MUST advertise the unsolicited transmission status for the requested Notification Type and MAY advertise the current state of one of its state variables within the Notification Type.

Pull nodes:

CC:0071.03.04.11.007 The value 0xFF indicates to the receiving node that it MUST retrieve the next Notification in its queue.

CC:0071.03.04.12.003 A Notification Type value indicates to the receiving node that it SHOULD retrieve the next Notification in its Notification queue matching the specified Notification Type value.

Notification Event / State (8 bits)

This field is used to optionally specify a Notification Event/State within the Notification Type. Assigned values are defined in [18]. This field allows receiving nodes to differentiate between V2 Alarm Get Command and V3 or newer Notification Get Command.

CC:0071.03.04.11.008 If the Notification Type is set to 0xFF, this field MUST be set to 0x00 by a sending node and SHOULD be ignored by a receiving node.

Push nodes:

CC:0071.03.04.13.001 A sending node MAY set this field to 0x00.

CC:0071.03.04.13.002 A sending node MAY specify a value that is supported by the receiving node within the specified Notification Type. In this case, a receiving node:

- MAY advertise a current state related to the indicated state
- MAY return the same value in response to a supported Notification Event.

CC:0071.03.04.11.009 A receiving node MUST return the “Unknown notification” value (0xFE) in response to a non-supported Notification Event.

Pull nodes:

CC:0071.03.04.12.004 This field SHOULD be ignored by a receiving node.

4.74.6 Notification Report Command

Push nodes:

This command is used for two purposes:

- If this command is returned in response to a Notification Get Command, this command advertises if the unsolicited transmission of the advertised Notification Type is enabled and optionally advertises a currently active state.
- If this command is sent unsolicited, it advertises an event or state Notification.

The use of the command’s fields is the same in both cases.

Pull nodes:

This command is used by a sending node to return a Notification from its queue or indicate that its queue is empty.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_NOTIFICATION							
Command = NOTIFICATION_REPORT							
V1 Alarm Type							
V1 Alarm Level							
Reserved							
Notification Status							
Notification Type							
Notification Event / State							
Sequence	Reserved		Event / State Parameters Length				
Event /State Parameter 1 (optional)							
...							
Event / State Parameter N (optional)							
Sequence Number (optional)							

Reserved

CC:0071.03.05.11.001 This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

V1 Alarm Type (8 bits) & V1 Alarm Level (8 bits)

These fields carry the proprietary Alarm Type and Alarm Level fields originally introduced with Alarm
CC:0071.03.05.11.002 Command Class, Version 1. V1 Alarm Type and V1 Alarm Level fields MUST be specified in the product manual.

CC:0071.03.05.11.003 If the V1 Alarm Type is not supported, these fields MUST be set to 0x00.

Notification Status (8 bits)

Push nodes:

This field is used to advertise the status of the Notification Type indicated in this command.

CC:0071.03.05.11.004 This field MUST comply with Table 93.

Table 93, Notification Report::Notification Status (push nodes)

Value	Description	Version
0x00	Unsolicited transmissions are disabled for the specified Notification Type	2
0xFF	Unsolicited transmissions are enabled for the specified Notification Type	2

All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

Pull nodes:

This field is used to advertise the status of the sending node's Notification queue.

CC:0071.03.05.11.005 This field MUST comply with Table 94

Table 94, Notification Report::Notification Status (pull nodes)

Value	Description	Version
0x00	This command carries a valid Notification returned from the queue. There may be more Notifications queued up. This Notification MAY be persistent.	2
0xFE	This command does not carry any valid Notification and the event queue is empty	3

All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

Notification Type (8 bits)

This field is used to advertise the Notification Type. Assigned values are defined in [18].

CC:0071.03.05.11.006 A sending node MUST set this field to the Notification Type of the actual Notification.

Notification Event / State (8 bits)

This field is used to specify a Notification Event/State for the advertised Notification Type. Assigned values are defined in [18].

CC:0071.03.05.11.007 A node advertising a Notification MUST set this field to the actual Notification event/state.

Sequence (1 bit)

This field is used to advertise the presence of the "Sequence Number" field.

CC:0071.03.05.11.008 The value 0 MUST indicate that no "Sequence Number" field is appended after the "Event / State Parameter" field.

CC:0071.03.05.11.009 The value 1 MUST indicate that a "Sequence Number" field is appended after the "Event / State Parameter" field.

Event / State Parameters Length (5 bits)

This field is used to advertise the length in bytes of the Event / State Parameters field.

CC:0071.03.05.11.00A The value 0 MUST indicate that no Event / State Parameter field is appended after the Event Parameters Length field.

CC:0071.03.05.11.00B Values in the range 1..31 MUST indicate the length of the Event / State Parameter field appended after the Event / State Parameters Length field.

Event / State Parameter (N bytes)

The Event / State Parameter field is used to specify associated parameters to a Notification.

Parameters for each Event/State Notification are defined in [18].

CC:0071.03.05.13.001
CC:0071.03.05.11.00C
This field MAY carry an encapsulated command. In this case, the field MUST include the complete command structure, i.e. Command Class, Command and all mandatory command fields.

CC:0071.03.05.11.00D
If a node encapsulates a command in this field, the corresponding Command Class MUST be supported by the node and advertised in the Node Information Frame (NIF) or S0/S2 Supported Command Report.

CC:0071.03.05.11.00E
This field MUST be omitted if the "Event Parameter Length" field is set to 0.

4.74.6.1 provides an example of Event Parameter encapsulation.

Sequence Number (8 bits)

CC:0071.03.05.13.002
This field is used to advertise a sequence number for the actual Notification. This command MAY carry a Sequence Number field.

CC:0071.03.05.11.00F
This field MUST be omitted if the Sequence flag is set to 0.

CC:0071.03.05.11.010
The first sequence number for each distinct Notification MUST be 1. A sending node MUST increment the sequence number for a Notification each time it issues a Notification Report for that given Notification.

CC:0071.03.05.11.011
The Sequence Number range MUST be in the range 0..255. The value after 255 MUST be 0.

Example:

1. Notification {Smoke Alarm::Smoke Detected}, …, Seq. Number = 254
2. Notification {Smoke Alarm::Smoke Detected}, …, Seq. Number = 255
3. Notification {Heat Alarm::Overheat Detected}, …, Seq. Number = 6
4. Notification {Smoke Alarm::Smoke Detected}, …, Seq. Number = 0
5. Etc.

4.74.6.1 Event / State parameter encapsulation

CC:0071.03.05.11.012
Event / State parameter encapsulation of commands MUST comprise the entire command, starting from the Command Class identifier.

For example, a "User Code Report" is used as Event / State Parameter in a "Keypad Lock/Unlock Operation". The "User Code Report" has the following format:

- Command Class =COMMAND_CLASS_USER_CODE = 0x63
- Command = USER_CODE_REPORT = 0x03
- User Identifier = 0x01
- User ID Status = 0x01
- User Code = 0x30, 0x30, 0x30, 0x30

The complete User Code Report therefore comprises the following Bytes: [0x63, 0x03, 0x01, 0x01, 0x30, 0x30, 0x30, 0x30].

Another example with the Node naming and location Command class is given in Table 95.

Table 95, Notification Report::Event / State parameter encapsulation (example)

Notification Report Command fields		Value	Explanation
1	COMMAND_CLASS_NOTIFICATION	0x71	Notification CC id
2	NOTIFICATION_REPORT	0x05	Notification Report command id
3	V1 Alarm Type	0x00	Not implemented
4	V1 Alarm Level	0x00	Not implemented
5	Reserved	0x00	Reserved field
6	Notification Status	0xFF	Unsolicited report is activated
7	Notification Type	0x01	Smoke Alarm
8	Notification Event	0x01	Smoke Detected
9	Sequence Number / Event Parameters Length	0x1A	Sequence Number is appended / Event Parm Length = 10
10	Event Parm 1	0x77	Node Naming & Location CC id
11	Event Parm 2	0x06	Node Location Report command id
12	Event Parm 3	0x00	Cmd Parm: Char = ASCII
13	Event Parm 4	0x4B	Cmd Parm: Node Location Char 1 = K
14	Event Parm 5	0x49	Cmd Parm: Node Location Char 2 = I
15	Event Parm 6	0x54	Cmd Parm: Node Location Char 3 = T
16	Event Parm 7	0x43	Cmd Parm: Node Location Char 4 = C
17	Event Parm 8	0x48	Cmd Parm: Node Location Char 5 = H
18	Event Parm 9	0x45	Cmd Parm: Node Location Char 6 = E
19	Event Parm 10	0x4E	Cmd Parm: Node Location Char 7 = N
20	Sequence Number	0x0F	Sequence Number = 15

4.74.7 Notification Supported Get Command

This command is used to request supported Notification Types.

CC:0071.03.07.11.001 The Notification Supported Report Command MUST be returned in response to this command.

CC:0071.03.07.11.002 This command MUST NOT be issued via multicast addressing.

CC:0071.03.07.11.003 A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_NOTIFICATION							
Command = NOTIFICATION_SUPPORTED_GET							

4.74.8 Notification Supported Report Command

This command is used to advertise supported Notification Types.

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)

This field is used to indicate if the node implements proprietary Alarms from version 1.

CC:0071.03.08.11.001 The value 0 MUST indicate that the device implements only Notification CC V2 or newer Notification Types.

CC:0071.03.08.11.002 The value 1 MUST indicate that the device implements Notification CC V2 Notification Types as well as proprietary Alarm CC V1 Alarm Types and Alarm Levels.

Number of Bit Masks (5 bits)

CC:0071.03.08.11.003 This field MUST advertise the length in bytes of the Bit Mask field.
The value MUST be in the range 1..31.

Bit Mask (N bytes)

CC:0071.03.08.11.004 The Bit Mask field describes the supported Notification Types by the node. The length of this field in bytes MUST match the value advertised in the Number of Bit Masks field.

- Bit 0 in Bit Mask 1 is not allocated to any Notification Type and MUST be set to zero.
- Bit 1 in Bit Mask 1 indicates if Notification Type = Smoke Alarm (0x01) is supported.
- Bit 2 in Bit Mask 1 indicates if Notification Type = CO Alarm (0x02) is supported.
- Bit 3 in Bit Mask 1 indicates if Notification Type = CO_2 Alarm (0x03) is supported
- …

For Notification Types, refer to [18].

CC:0071.03.08.11.005 If the Notification Type is supported, the corresponding bit MUST be set to 1.
If the Notification Type is not supported, the corresponding bit MUST be set to 0.

CC:0071.03.08.11.006 The Notification Type values 0x00 and 0xFF are special-purpose values. Reserved values and special-purpose values MUST NOT be advertised in the Bit Mask field by a sending node and MUST be ignored by a receiving node.

4.74.9 Event Supported Get Command

This command is used to request the supported Notifications for a specified Notification Type.

CC:0071.03.01.11.001 The Event Supported Report Command MUST be returned in response to this command.

CC:0071.03.01.11.002 This command MUST NOT be issued via multicast addressing.
CC:0071.03.01.11.003 A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_NOTIFICATION							
Command = EVENT_SUPPORTED_GET							
Notification Type							

Notification Type (8 bits)

For Notifications Types, refer to [18].

CC:0071.03.01.11.004 If a node receives an unsupported Notification Type or a special-purpose value, the receiving node MUST respond with Event Supported Report Command with the Notification Type specified in the this command and the Number of Bit Masks field set to 0.

4.74.10 Event Supported Report Command

This command is used to advertise supported events/states for a specified Notification Type.

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 bits)

For Notifications Types, refer to [18].

Number of Bit Masks (5 bits)

This field is used to advertise the length (in bytes) of the Bit Mask field.

CC:0071.03.02.11.001 The value MUST be in the range 0..31.
The value 0 MUST indicate that the Notification Type is not supported.

Bit Mask (N bytes)

This field is used to advertise the supported Events/States for the actual advertised Notification Type.
CC:0071.03.02.11.002 The length of this field (in bytes) MUST match the value advertised in the Number of Bit Masks field.
This field MUST be omitted if the “Number of Bit Masks” field is set to 0.

CC:0071.03.02.11.003 The bit value ‘1’ MUST indicate that the actual Event/State is supported.
The bit value ‘0’ MUST indicate that the actual Event/State is not supported.

Example: Notification Type = Heat Alarm (0x04):

- Bit 0 in Bit Mask 1 field is not allocated to any event and must therefore be set to zero.
- Bit 1 in Bit Mask 1 field indicates support for Overheat detected (0x01).
- Bit 2 in Bit Mask 1 field indicates support for Overheat, unknown loc. (0x02).
- Bit 3 in Bit Mask 1 field indicates support for Rapid temp rise (0x03).
- …

For Notification Types and their Events/States, refer to [18].

CC:0071.03.02.11.004 The Event/State values 0x00 and 0xFE are special-purpose values. Reserved values and special-purpose values MUST NOT be advertised in the Bit Mask field by a sending node and MUST be ignored by a receiving node.

4.75 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

4.75.1 Prepayment Balance Get Command

This command is used to request the balance of the Prepayment.

The Prepayment Balance Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_PREPAYMENT							
Command = PREPAYMENT_BALANCE_GET							
Balance Type			Reserved				

Balance Type (2 bits)

The field specifies which balance type is requested in the response report. The available balance types may be requested using the Prepayment Supported Get Command.

Table 96, Prepayment Balance Get::Balance Type encoding

Value	Balance Type
0x00	Utility Balance
0x01	Monetary Balance

All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

4.75.2 Prepayment Balance Report Command

This command is used to report the current balances.

The report includes the following main elements:

- Balance
- Debt
- Emergency Credit

The elements MAY be given in monetary values or in utility units depending on the Balance Type field.

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 bits)

The field specifies which type of balance is given in the report.

All available Balance Types may be found in the section 4.75.1 Prepayment Balance Get Command.

Meter Type (6 bits)

Meter Type specifies the type of metering device the command originates. This field MUST be encoded according to the Meter Types defined in [21].

Scale (5 bits)

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 4.62.13 Meter Table Current Data Report Command.

Currency (3 bytes)

This field advertises the currency code. Reports of the type “Monetary Balance” MUST advertise currency codes complying with ISO 4217. Other report types MUST advertise a currency code of “XXX”.

Table 97, Prepayment Balance Report::Currency examples

Currency Code	Currency 1	Currency 2	Currency 3
Pound sterling (ISO 4217)	G	B	P
US Dollar (ISO 4217)	U	S	D
No Currency	X	X	X

Balance, Debt and Emergency Credit 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.

Balance, Debt and Emergency Credit (32 bits)

The Balance, Debt and Emergency Credit fields MUST be encoded as 32-bit signed integers. The first byte MUST carry most significant byte. Table 10, Signed field encoding (two’s complement representation) shows signed decimal values together with their hexadecimal equivalents.

Debt Recovery Percentage (8 bits)

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).

4.75.3 Prepayment Supported Get Command

The Prepayment Supported Get Command is used to request type of Balance Reports that are available in the device.

The Prepayment Supported Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_PREPAYMENT							
Command = PREPAYMENT_SUPPORTED_GET							

4.75.4 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 bits)

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 may be found in the section 4.75.1 Prepayment Balance Get Command.

4.76 Prepayment Encapsulation Command Class, version 1

The Prepayment Encapsulation Command Class is used to smartcard preinstalled security mechanisms.

4.76.1 Prepayment Encapsulation Command

This 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 transparently to Z-Wave.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_PREPAYMENT_ENCAPSULATION							
Command = CMD_ENCAPSULATION							
Data 1							
...							
Data N							

Data (N bytes)

This field contains prepayment smart card data.

4.77 Proprietary Command Class, version 1 [DEPRECATED]

THIS COMMAND CLASS HAS BEEN DEPRECATED

A device MAY implement this command class, but it is RECOMMENDED that new implementations use the Manufacturer Proprietary Command Class.
If implementing this command class, it is RECOMMENDED that the Manufacturer Proprietary Command Class is also implemented.

The Proprietary Command Class is used to transfer data between devices. The data content MUST be vendor specific and commands MUST NOT provide any value-add with respect to the Home Automation application in general.

4.77.1 Proprietary Set Command

This command is 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 (N bytes)

The data fields may be used to set various data in the device. The number of data fields transmitted MUST be determined from the length field in the frame.

4.77.2 Proprietary Get Command

This command is used to request data from a device.

The Proprietary Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_PROPRIETARY							
Command = PROPRIETARY_GET							
Data 1							
…							
Data N							

Data (N bytes)

Refer to explanation under the Proprietary Set Command.

4.77.3 Proprietary Report Command

This command is used to retrieve various data from a device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_PROPRIETARY							
Command = PROPRIETARY_REPORT							
Data 1							
…							
Data N							

Data (N bytes)

Refer to explanation under the Proprietary Set Command.

4.78 Protection Command Class, version 1

The Protection Command Class version 1 used to protect a device against unintentional control by e.g. a child.

4.78.1 Protection Set Command

This command is 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							
Protection State							

Protection State (8 bits)

The Protection State field used to set the protection state of the device.

Table 98, Protection Set::Protection State encoding

Protection State	Description
0x00	Unprotected - The device is not protected, and may 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.

All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

Control via Z-Wave is always possible independently of the protection state.

4.78.2 Protection Get Command

This Command is used to request the protection state from a device.

The Protection Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_PROTECTION							
Command = PROTECTION_GET							

4.78.3 Protection Report Command

This command is used to report the protection state of a device.

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.

4.79 Protection Command Class, version 2

The Protection Command Class version 2 is extended to specify whether a device may 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 off when it is recording her/his favorite show. In this case the Protection Command Class version 2 may 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 remain unchanged**.**

This Command Class is intended for convenience applications. The Command Class SHOULD NOT be used for safety critical applications.

4.79.1 Protection Set Command

This command is 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 Local Protection State field used to set the protection state of the device.

Table 99, Protection Set::Local Protection State encoding

Local Protection State	Description
0	Unprotected - The device is not protected, and may 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.

All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.
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)

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 the Application Rejected Request Command (Status = 0) of the Application Status Command Class. Refer to the Application Status Command Class.

Table 100, Protection Set::RF Protection State

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.

All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.**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.

4.79.2 Protection Report Command

This command is used to report the protection state of a device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_PROTECTION							
Command = PROTECTION_REPORT							
Reserved				Local Protection State			
Reserved				RF Protection State			

For field description, refer to 4.79.1 Protection Set Command.

4.79.3 Protection Supported Get Command

This command is used to query supported protection capabilities.

The Protection Supported Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_PROTECTION							
Command = PROTECTION_SUPPORTED_GET							

4.79.4 Protection Supported Report Command

This command is used to advertise supported protection capabilities.

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 (2 bytes)

The list of all Local Protection States may be found in section 4.79.1. 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 (2 bytes)

The list of all RF Protection States may be found in section 4.79.1. 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 (1 bit)

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 (1 bit)

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.

4.79.5 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.

4.79.5.1 Protection Exclusive Control Set Command

This command is used to set the NodeID 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							
NodeID							

NodeID

The NodeID 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. In that case, the Application Rejected Request Command MUST be returned.

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 NodeID for exclusive control MUST be set to 0. To reset the exclusive control state in a device an Exclusive Control Set Command with NodeID 0 as parameter MUST be send to the device.

4.79.5.2 Protection Exclusive Control Get Command

This command is used to request a Protection Exclusive Control Report Command from the device.

The Protection Exclusive Control Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_PROTECTION							
Command = PROTECTION_EC_GET							

4.79.5.3 Protection Exclusive Control Report Command

This command is used to return the NodeID 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							
NodeID							

NodeID

See description under the Protection Exclusive Control Set Command section 4.79.5.1.

4.79.6 Protection Timeout

The Protection Timeout is an optional feature. The Commands in this section MAY be implemented if the device supporting Protection Command Class version 2 announces support for Timeout in the Protection Supported Report Command.

4.79.6.1 Protection Timeout Set Command

This command is 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 describes the time that a device MUST remain in RF Protection mode.
Factory default setting for the Timeout parameter MUST be 0x00. This field MUST be encoded according to Table 101.

Table 101, Protection Timeout Set::Timeout encoding

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.

All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

4.79.6.2 Protection Timeout Get Command

This command is used to request a Protection Timeout Report Command from the device.

The Protection Timeout Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_PROTECTION							
Command = PROTECTION_TIMEOUT_GET							

4.79.6.3 Protection Timeout Report Command

This command is 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

This field indicates the remaining timeout set in the Node. It MUST be encoded as described in Table 101

4.80 Pulse Meter Command Class, version 1 [DEPRECATED]

THIS COMMAND CLASS HAS BEEN DEPRECATED

A device MAY implement this command class, but it is RECOMMENDED that new implementations use the Meter Command Class.
If implementing this command class, it is RECOMMENDED that the Meter Command Class is also implemented.

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.

4.80.1 Pulse Meter Get Command

This command is used to request the number of pulses that has been counted.

The Pulse Meter Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER_PULSE							
Command = METER_PULSE_GET							

4.80.2 Pulse Meter Report Command

This command is used to report the number of pulses detected.

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.

4.81 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.

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.

4.81.1 Rate Table Set Command

This command adds a *rate parameter set* to a given *rate parameter set identifier*.

<table>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="8">Command Class = COMMAND_CLASS_RATE_TBL_CONFIG</td></tr>
<tr><td colspan="8">Command = RATE_TBL_SET</td></tr>
<tr><td colspan="8">Rate Parameter Set ID</td></tr>
<tr><td>Reserved</td><td colspan="2">Rate Type</td><td colspan="5">Number of Rate Char.</td></tr>
<tr><td colspan="8">Rate Character 1</td></tr>
<tr><td colspan="8">…</td></tr>
<tr><td colspan="8">Rate Character N</td></tr>
<tr><td colspan="8">Start Hour Local Time</td></tr>
<tr><td colspan="8">Start Minute Local Time</td></tr>
<tr><td colspan="8">Duration Minute 1</td></tr>
<tr><td colspan="8">Duration Minute 2</td></tr>
</table>

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.

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 bits)

The Rate Parameter Set ID indicates the requested parameter set.

Rate Type (2 bits)

Rate Type specifies the type of parameters in the report. This field MUST be encoded according to Table 69.

Reserved

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Number of Rate Characters (5 bits)

Number of characters defining the rate text (1..32).

Rate Character (N bytes)

The Rate character fields hold the string identifying the rate parameter set. The character presentation uses standard ASCII codes (values in the range 128..255 MUST be ignored).

Start Hour Local Time (8 bits)

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

Start 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.

Duration Minute (16 bits)

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 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.

Consumption Scale (5 bits)

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 4.62.13 Meter Table Current Data Report Command.

Minimum / Maximum Consumption Value (4 * 8 bits)

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 show the value even though the Scale is not supported.

Max. Demand Precision (3 bits)

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 bits)

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 4.62.13 Meter Table Current Data Report Command.

Max. Demand Value (4 * 8 bits)

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.

NOTICE: The device receiving the Rate Table Report MUST show the value even though the Scale is not supported.

DCP Rate ID (8 bits)

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.

4.81.2 Rate Table Remove Command

This 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

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Rate Parameter Set IDs (6 bits)

The Rate Parameter Set IDs indicates the number of rate parameter set IDs in the command.

Rate Parameter Set ID (N bytes)

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.

4.82 Rate Table Monitor Command Class, version 1

The Rate Table Monitor Command Class defines the parameter sets for a range of rates.

4.82.1 Rate Table Supported Get Command

This command is used to request the number of rates and parameter sets supported by the Rate Table Command Class.

The Rate Table Supported Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_RATE_TBL_MONITOR							
Command = RATE_TBL_SUPPORTED_GET							

4.82.2 Rate Table Supported Report Command

This command is used to advertise the number of rates and parameter sets supported.

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 bits)

Number of rates supported (1..255).

Parameter Set Supported Bit Mask 1, 2 (16 bits)

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:

Table 102, Rate Table Supported Report::Parameter Set Supported Bit Mask encoding

Parameter Set Supported	Bit Map	Description
1	Bit 0	Reserved
1	Bit 1	Supports Block tariffs 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 Maximum demand tariffs 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 Subscribed demand tariffs 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.

All other bits are reserved and MUST be set to zero by a sending node. Reserved bits MUST be ignored by a receiving node.

4.82.3 Rate Table Get Command

This command is used to request the rate parameter set for a given rate parameter set identifier.

The Rate Table Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

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 bits)

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.

4.82.4 Rate Table Report Command

This command reports rate parameter set for a given rate parameter set identifier.

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.

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 4.81.1).

4.82.5 Rate Table Active Rate Get Command

This command is used to retrieve the rate currently active in the meter.

The Rate Table Active Rate Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_RATE_TBL_MONITOR							
Command = RATE_TBL_ACTIVE_RATE _GET							

4.82.6 Rate Table Active Rate Report Command

This command is used to advertise 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 bits)

The Rate Parameter Set ID of the rate currently active in the meter.

4.82.7 Rate Table Current Data Get Command

This command is used to request a number of time stamped values (current) in physical units according to the dataset mask.

The Rate Table Current Data Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

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 bits)

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 (24 bits)

The dataset requested parameter indicates which data is requested by the command. This field MUST be encoded according to the Meter Datasets defined in [21].

4.82.8 Rate Table Current Data Report Command

This command is used to report a number of time stamped values (current) in physical units in the device.

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 bits)

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.

Rate Parameter Set ID (8 bits)

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 (24 bits)

The dataset parameter indicates which data is included in the report. This field MUST be encoded according to the Meter Dataset defined in [21].

Year (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). A year equal to 0x0000 indicates that a accumulated value is not determined yet.

Day (8 bits)

Specify the day of the month between 01 and 31.

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.

Precision (N * 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.

Current Scale (N*5 bits)

The Current Scale is used to indicate the scale (unit) of the following value. This field MUST be encoded according to the Meter Dataset and Scales defined in [21].

Current Value (N * 32 bits)

The Current Value advertises a value corresponding to the Dataset Requested field of the Get Command. The field MUST be encoded as a 32-bit signed integer. The first byte (Value 1) MUST carry most significant byte. Table 10, Signed field encoding (two's complement representation) shows signed decimal values together with their hexadecimal equivalents.

NOTICE: The device receiving the Rate Table Current Data Report MUST show the value even though the Scale is not supported.

4.82.9 Rate Table Historical Data Get Command

This command is used to request a number of time stamped values (historical) in physical units according to rate type, dataset mask and time interval.

The Rate Table Historical Data Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

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 bits)

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 bits)

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 (24 bits)

The dataset requested parameters indicate data requested. This field MUST be encoded according to the Meter Dataset defined in [21].

Start/Stop Year (16 bits)

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

Start/Stop Month (8 bits)

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 bits)

Specify the day of the month between 01 and 31.

Start/Stop Hour Local Time (8 bits)

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

Start/Stop 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.

Start/Stop 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.

4.82.10 Rate Table Historical Data Report Command

This command is used to report a number of time stamped values (historical) in physical units in the device.

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 bits)

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.

Rate Parameter Set ID (8 bits)

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 (24 bits)

The dataset parameter indicates which data is included in the report. This field MUST be encoded according to the Meter Dataset defined in [21].

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). A year equal to 0x0000 indicates that a accumulated value is not determined yet.

Day (8 bits)

Specify the day of the month between 01 and 31.

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.

Historical Precision (N * 3 bits)

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 (N * 5 bits)

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 4.62.13 Meter Table Current Data Report Command.

Historical Value (N * 32 bits)

The Historical Value is a 32 bit signed field defined by dataset requested field. The first byte (Value 1) is the most significant byte. shows signed decimal values together with their hexadecimal equivalents.

NOTICE: The device receiving the Rate Table Historical Data Report MUST show the value even though the Scale is not supported.

4.83 Scene Activation Command Class, version 1

The Scene Activation Command Class used for launching scenes in a number of actuator nodes e.g. another scene-controlling unit, a multilevel switch, a binary switch etc.

4.83.1 Compatibility considerations

A node supporting this Command Class MUST also support the Scene Actuator Configuration Command Class.

This command class requires an initial configuration of the scenes to be launched by the Scene Actuator Configuration Set.

Since a common identifier that is sent out, multicast addressing may be used. Multicast addressing may eliminate the potential popping effect which could be the result if individual Set Commands were send out to a large number of nodes distributed over a vast area. The multicast transmission MUST be followed by individual singlecast transmissions to ensure all the nodes have received the command.

4.83.2 Scene Activation Set Command

This command is used to activate the setting associated to 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)

This field is used to specify the Scene ID that the receiving node MUST activate

This field MUST be in the range 1..255.

Dimming Duration (8 bits)

This field MUST specify the time that the transition from the current level to the target level (indicated by the Scene ID) should take.

Supporting actuator nodes without duration capabilities MUST ignore this field.

Supporting actuator nodes with a duration capabilities SHOULD respect the indicated dimming duration to reach the target level.

This field MUST be encoded according to Table 103.

Table 103, Scene Activation Set:: Dimming Duration encoding

Value	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.

4.84 Scene Actuator Configuration Command Class, version 1

The Scene Actuator Configuration Command Class is used to configure scenes settings for a node supporting an actuator Command Class, e.g. a multilevel switch, binary switch etc.

4.84.1 Compatibility considerations

A node supporting this Command Class MUST support 255 Scene IDs (Scene ID 1 to 255). The Scene ID 0 MUST NOT be supported..

4.84.2 Scene Actuator Configuration Set Command

This command is used to associate the specified scene ID to the defined actuator 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)

This field is used to specify the Scene ID to be associated with the current actuator settings.

This field MUST be in the range 1..255.

Dimming Duration (8 bits)

This field MUST specify the time it MUST take to reach the target level associated to the actual Scene ID.

A receiving node always starts from current level. The dimming duration MUST be the same independently of the number of levels to be changed.

Supporting actuator nodes without duration capabilities MUST ignore this field.

Supporting actuator nodes with duration capabilities SHOULD respect the indicated duration to reach the target level.

This field MUST be encoded according to Table 7, Actuator Command Class Duration Set encoding.

Override (1 bit)

This field is used to specify if the current actuator settings MUST be used as settings for the actual Scene ID.

If this field is set to 0, the current actuator settings at the supporting nodeMUST be associated to the actual Scene ID. In this case, the Level field MUST be ignored.

If this field is set to 1, the value indicated by the Level field MUST be associated to the actual Scene ID.

Reserved

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Level (8 bits)

This field is used to specify the actuator setting that MUST be used as setting for the actual Scene ID. This field MUST correspond and be interpreted as the Value field of the Basic Set Command. i.e. the supporting node receiving a Scene Activation Set for the actual Scene ID specified in this command MUST react as if it had received a Basic Set Command with the Value field set to the value indicated by this field.

4.84.3 Scene Actuator Configuration Get Command

This command is used to request the settings for a given scene identifier or for the scene currently active.

The Scene Actuator Configuration Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

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)

This field is used to specify the requested Scene ID settings.

Values in the range 1..255 MUST indicate that the receiving node MUST return settings associated to the actual Scene ID..

The value 0 MUST indicate that the current active scene (if any).

4.84.4 Scene Actuator Configuration Report Command

This command is used to advertise the settings associated to a scene identifier.

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)

This field is used to specify the actual Scene ID for which the settings are being advertised.

Values in the range 1..255 MUST indicate an actual Scene ID.

The value 0 MUST indicate that no scene is currently active at the sending node and no scene setting is advertised in this command. In this case, the Level and Dimming Duration fields SHOULD be ignored.

Level (8 bits)

This field is used to advertise the actuator setting that MUST be used by the node when activating the actual Scene ID.

The value advertised by the sending node MUST correspond to the format of the Value field when returning a Basic Report Command.

Dimming Duration (8 bits)

This field MUST specify the time it MUST take to reach the target level associated to the actual Scene ID..

Supporting actuator nodes without duration capabilities MUST ignore this field and SHOULD set it to 0x00.

Supporting actuator nodes with duration capabilities SHOULD respect the indicated duration to reach the target level.

This field MUST be encoded according to Table 104.

Table 104, Scene Actuator Configuration Report::Dimming Duration encoding

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.

All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

4.85 Scene Controller Configuration Command Class, version 1

The Scene Controller Configuration Command Class is used to configure nodes launching scenes using their association groups

4.85.1 Compatibility Considerations

A node supporting this Command Class MUST support 255 scene IDs (Scene ID 1 to 255). The Scene ID 0 MUST NOT be supported..

A node supporting this Command Class MUST support the Association Command Class, version 1 or newer.

A node supporting this Command Class MUST issue Scene Activation Set Command via its association groups.

4.85.2 Scene Controller Configuration Set Command

This command is 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)

This field is used to specify an Association Group Identifier.

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 may be used to request the number of groupings that a node supports.

A sending node SHOULD NOT specify Group ID 1 as it is reserved for the Lifeline association group.

Scene ID (8 bits)

This field is used to specify the Scene ID that MUST be associated with the actual grouping identifier.

Values in the range 1..255 MUST indicate that the receiving node MUST associate the Scene ID and Dimming Duration values to the actual Association Group ID. The supporting node MUST set the Scene ID value indicated by this field in the Scene Activation Set Command when issuing it via the actual Group ID.

The value 0 MUST indicate that the receiving node MUST disable an associated scene for the specified Group ID.

Dimming Duration (8 bits)

This field is used to specify what value the supporting node MUST set in the Dimming Duration field of the Scene Activation Set Command when issuing it via the actual Group ID.

4.85.3 Scene Controller Configuration Get Command

This command is used to request the settings for a given association grouping identifier or the active settings.

The Scene Controller Configuration Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

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)

This field is used to request scene settings configured for an actual Association Group Identifier.

Values in the range 1..255 MUST indicate the requested Group Identifier.

The value 0 MUST indicate that the receiving node MUST return the currently active Group Identifier and Scene ID (last activated group/scene).

A node receiving a non-supported group or having no currently active Group Identifier SHOULD return a report for Group ID 0.

4.85.4 Scene Controller Configuration Report Command

This command is used to advertise the current scene controller settings.

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)

This field is used to advertise the actual Association Group Identifier for which the settings are being advertised.

Scene ID (8 bits)

This field is used to specify the Scene ID that is associated with the actual grouping identifier.

Value in the range 1..255 MUST indicate the actual Scene ID setting for the specified grouping identifier.

The value 0 MUST indicate that the Group ID/Scene ID is disabled.

Dimming Duration (8 bits)

This field is used to specify the Dimming Duration that is associated with the actual grouping identifier.

If the actual Group ID/Scene ID is disabled, a sending node SHOULD set this field to 0.

4.86 Schedule Command Class, version 1

The Schedule Command Class allows scheduling the execution of commands for a given duration in a supporting device. It is a generic Command Class that may be used to schedule commands of any other Command Class.

4.86.1 Terminology

A schedule or regular schedule is a delayed execution of one or more commands for a given duration. The commands used in a schedule are typically "Set" type commands, affecting actuating resources or states.

A schedule is first created or set, meaning that an available Schedule ID is used and has been assigned a set of commands, a starting time and a duration.

A schedule can be removed, meaning that a previously set Schedule ID is freed and the assigned commands, starting time and duration are erased.

A schedule is active during the configured duration after its starting time.
When a schedule is active, the states affected by the schedule commands are temporarily changed. When a schedule becomes inactive, the states affected by the schedule commands are restored to their previous values.

Several schedules can overlap and be active simultaneously.

Supporting nodes can optionally support enabling and disabling schedules. By default, a schedule is enabled when being created. When a schedule is disabled, it will never become active, even if the start time condition is met. It allows a controlling node to quickly enable/disable schedules without erasing the schedules configuration from the supporting nodes.

When all schedules are inactive, the supporting node is said to be operating in Normal Mode. Regardless of whether any schedule is active, Normal Mode's states are permanently affected by direct commands (issued without Schedule encapsulation). State changes triggered by schedules do not affect Normal Mode.

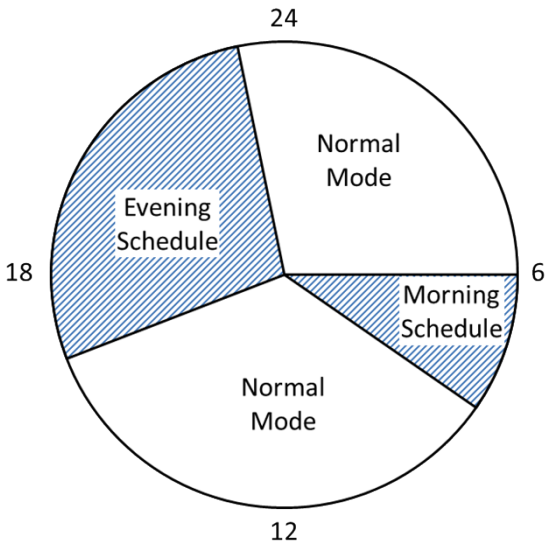


Figure 25. Simple daily schedules (example)

Direct commands affecting the states of a currently active schedule will cause those states to go back to normal mode and the corresponding normal mode states/values to be permanently changed.

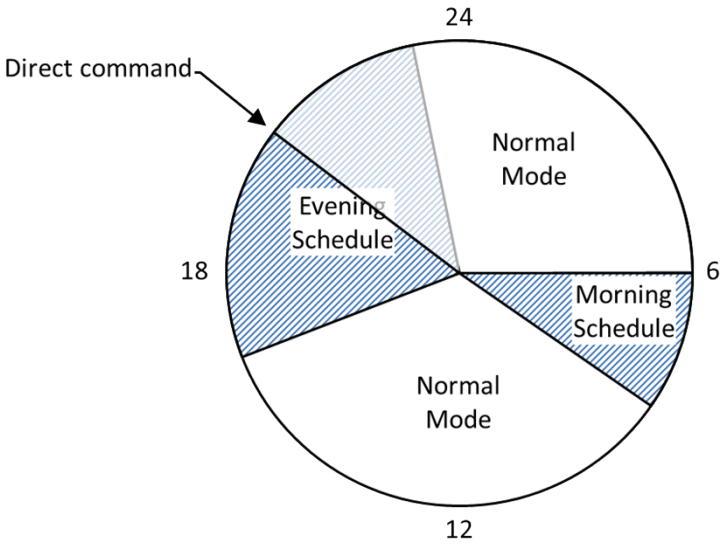


Figure 26. Receiving a direct command during daily schedules (example)

There are 2 types of special schedules.

If a Fall Back Schedule is created, it takes over the role of the Normal Mode. The Fall Back Schedule is activated when all other schedules are inactive. Fall Back Schedule is used to define default settings or states for the normal mode

If an Override Schedule is created, it suspends all other schedules. An override schedule may have a start time or start immediately. An override schedule may have a duration, run until stopped or run until another regular schedule starts. An example is given in the figure below

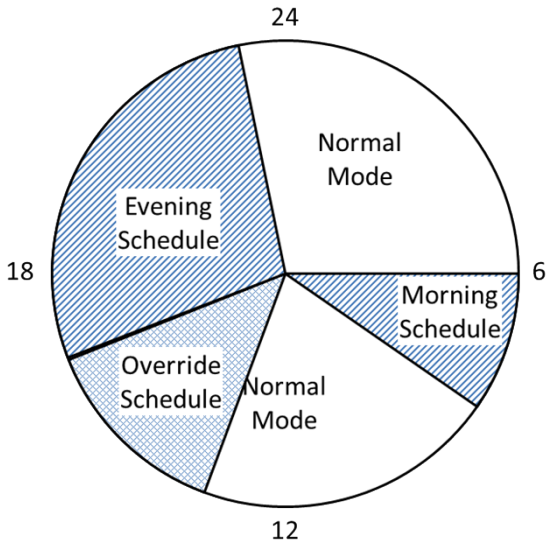


Figure 27. Daily schedules and an "Advance" Override Schedule (example)

An overview of the Schedule types and priorities is given in Table 105

Table 105. Schedule CC terminology and priority

Priority	Term	Description
(highest)	Direct command	Affects device state immediately. Permanently affects Normal Mode.
(higher)	Override Schedule (Schedule ID = 0xFF)	All other schedules are suspended as long as the override schedule is active.
(normal)	Regular Schedule(s) (Schedule ID in [1..Supported number of schedules])	One or more schedules defining intended behavior.
(lower)	Fall Back Schedule (Schedule ID = 0xFE)	Fall Back Schedule becomes active when no other schedule is currently active. If the Fall Back Schedule is defined, Normal Mode is never reached, unless receiving direct commands
(lowest)	Normal Mode	No schedules are currently active.

4.86.2 Interoperability considerations

CC:0053.01.00.31.001 A schedule causes a temporary state change that only applies to the duration of the schedule. The state value(s) of the Normal Mode MUST NOT be affected by any command executed as part of a schedule.

CC:0053.01.00.31.002 The state value(s) of the Normal Mode MUST be restored when the affecting Schedule become inactive.

CC:0053.01.00.32.001 Schedules are set with a start time, which requires the supporting node to be aware of the current date and/or time. It is RECOMMENDED to support the Clock Command Class or the Time Command Class in order to allow a controlling node to verify or configure correct date/time settings at the supporting node.

4.86.2.1 Override Schedule

CC:0053.01.00.31.003 When the Override Schedule is active, all regular schedules MUST be suspended.

CC:0053.01.00.31.004 When the Override Schedule ends, all suspended schedules MUST resume the state they would have had if the Override Schedule had not been activated.

CC:0053.01.00.32.002 If the Override Schedule is created with a duration type not set to Override, a receiving node SHOULD respect the specified duration for the Override Schedule.

CC:0053.01.00.31.005 A supporting node MUST respect the specified duration for all regular schedules.

4.86.2.2 Fall Back Schedule

CC:0053.01.00.31.006 If no Fall Back Schedule is set, a supporting node MUST return to Normal Mode when no schedule is currently active.

CC:0053.01.00.31.007 If a Fall Back Schedule is set, a supporting node MUST:

- Activate the Fall Back Schedule when no other schedule is active.
- De-activate the Fall Back Schedule if any other schedule becomes active.
- Direct commands state changes MUST be active until the start of a new schedule affecting the same state values.

4.86.2.3 Multi Channel considerations

A Multi Channel device may support different schedule types and command classes for each Multi Channel End Point. An example is a central heating boiler with a thermostat for room heating and another thermostat for water heating. Each system may implement regular weekday+time schedules as well as an override schedule for manual intervention.

CC:0053.01.00.32.003 In any case, it is RECOMMENDED that multi-function devices support the Multi Channel Command Class, so that each End Point supports its own scheduling functionality.

4.86.3 Schedule Supported Get Command

This command is used to query the schedule functionalities supported by a node.

CC:0053.01.01.11.001 The Schedule Supported Report Command MUST be returned in response to this command.

CC:0053.01.01.11.002 This command MUST NOT be issued via multicast addressing.

CC:0053.01.01.11.003 A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCHEDULE							
Command = SCHEDULE_SUPPORTED_GET							

4.86.4 Schedule Supported Report Command

This command is used to advertise the schedule functionalities supported by a node.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCHEDULE							
Command = SCHEDULE_SUPPORTED_REPORT							
Number of Supported Schedule IDs							
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
Override Support	Supported Override Types						

Number of Supported Schedule IDs (8 bits)

CC:0053.01.02.11.001 This field MUST advertise the number of regular Schedule IDs supported by the node.

CC:0053.01.02.11.002 The implemented Schedule IDs MUST be in the range 1..[Number of Supported Schedule IDs]. i.e. a node supporting 10 regular schedules MUST accept Schedule ID values in the range 1..10.

CC:0053.01.02.13.001 The following special Schedule IDs MAY also be supported by a device:

- Schedule ID = 0xFF (Override Schedule)
- Schedule ID = 0xFE (Fall Back schedule)

CC:0053.01.02.11.003 Special Schedule IDs MUST NOT be included in the number advertised in this field.

Start Time Support (6 bits)

This field is used to advertise the start time options supported for regular schedules by a node.

CC:0053.01.02.11.004 Regular schedules MUST support the advertised start time options.
CC:0053.01.02.12.001 The Override Schedule SHOULD support the advertised start time options.
CC:0053.01.02.11.005 This field MUST be treated as a bitmask and MUST be encoded according to Table 106

Table 106. Start Time Support encoding

Bit	Indicates support for	Version
0	Start now. Refer to 4.86.4.1	1
1	Start Hour and Minute. Refer to 4.86.4.2	1
2	Calendar time. Refer to 4.86.4.3	1
3	Weekdays. Refer to 4.86.4.4	1
4..5	Reserved	N/A

Each bit indicates the support for a given Start Time option.

CC:0053.01.02.11.006 The value 1 MUST indicate that the node supports the corresponding start time option.
CC:0053.01.02.11.007 The value 0 MUST indicate that the node does not support the corresponding start time option.

While support is advertised as a bitmask, the actual functionality is triggered by different combinations of Schedule Set start time fields. The following subsections outline the mandatory supported combinations when advertising support for the corresponding option.

4.86.4.1 Start Time option: Start now

The *start now* option is used to make a configured schedule start immediately (upon reception of the Schedule Set Command).

CC:0053.01.02.11.008 A node that supports the *start now* option, MUST support the creation of a schedule starting immediately when all start time fields are left unspecified.

4.86.4.2 Start Time option: Hour and Minute

The start hour and minute option is used to make a schedule start at a specified time of the day.

CC:0053.01.02.11.009 If Start Hour and Start Minute are specified, node supporting this option MUST activate the schedule at the specified start time.

CC:0053.01.02.11.00A A node supporting the *Hour and Minute* start option MUST support the creation of a schedule with the following start time fields' combination:

- Every day @ Hour:Minute
  (YYMMDD = 0xFF, 0x00, 0x00, Weekdays = 0x00, HH:MM = **Hour:Minute**)

CC:0053.01.02.11.00B A node that also supports the *Weekdays* start option MUST support the creation of a schedule with the following start time fields' combination:

- Same weekday(s) every week @ Hour:Minute
  (YYMMDD = 0xFF, 0x00, 0x00, Weekdays = **weekdays**, HH:MM = **Hour:Minute**)

CC:0053.01.02.11.00C A node that also supports the *Calendar Time* start option MUST support the creation of a schedule with the following start time fields' combination:

- Same day every month @ Hour:Minute
  (YYMMDD = 0xFF, 0x00, **day**, Weekdays = 0x00, HH:MM = **Hour:Minute**)
- Same day every year @ Hour:Minute
  (YYMMDD = 0xFF, **month**, **day**, Weekdays = 0x00, HH:MM = **Hour:Minute**)
- One specific date in a specific year @ Hour:Minute
  (YYMMDD = **year**, **month**, **day**, Weekdays = 0x00, HH:MM = **Hour:Minute**)

4.86.4.3 Start Time option: Calendar time

The calendar time option is used to make a schedule start at a specified date.

CC:0053.01.02.11.00D If Start Year, Start Month and Start Day are specified, node supporting this option MUST activate the schedule at the specified date(s).

CC:0053.01.02.11.00E A node that supports the *Calendar Time* option MUST support the creation of a schedule with the following start time fields' combination:

- Same day every month @ 00:00
  (YYMMDD = 0xFF, 0x00, **day**, Weekdays = 0x00, HH:MM = 0x1F, 0x3F)
- Same day every year @ 00:00
  (YYMMDD = 0xFF, **month**, **day**, Weekdays = 0x00, HH:MM = 0x1F, 0x3F)
- One specific date in a specific year @ 00:00
  (YYMMDD = **year**, **month**, **day**, Weekdays = 0x00, HH:MM = 0x1F, 0x3F)

4.86.4.4 Start Time option: Weekday

The start weekday option is used to make a schedule start at on specified weekdays.

CC:0053.01.02.11.00F If weekdays are specified, node supporting this option MUST activate the schedule on the specified weekdays.

CC:0053.01.02.11.010 A node that supports the *Weekdays* option MUST support the creation of a schedule with the following start time fields' combination:

- Same weekday(s) every week @ 00:00
  (YYMMDD = 0xFF, 0x00, 0x00, Weekdays = **weekdays**, HH:MM = 0x1F, 0x3F)

Fall Back Support (1 bit)

This bit is used to advertise the support of the Fall Back Schedule (Schedule ID = 0xFE).

CC:0053.01.02.11.011 The bit MUST be set to 1 if the node supports the Fall Back Schedule.

CC:0053.01.02.11.012 The bit MUST be set to 0 if the node does not support the Fall Back Schedule.

Support Enable/Disable (1 bit)

This bit is used to advertise the support for enabling/disabling set/used schedules via the Schedule State Set Command.

CC:0053.01.02.11.013 The bit MUST be set to 0 if the node supports enabling/disabling schedules.

CC:0053.01.02.11.014 The bit MUST be set to 1 if the node does not support enabling/disabling schedules.

CC:0053.01.02.11.015 A node that does not support enabling/disabling schedules MUST consider every set/used schedule ID as enabled.

Number of supported CC (8 bits)

CC:0053.01.02.11.016 This field MUST advertise the number of supported Command Classes advertised in this command with the Supported CC and Supported Command fields.

CC:0053.01.02.11.017 The value 0xFF MUST be used if all the command classes supported by the node are also supported for scheduling. If this field is set to 0xFF:

CC:0053.01.02.11.018
- This command MUST NOT carry any Command Class entries (Supported CC and Supported Command fields).

CC:0053.01.02.11.019
- The node MUST support “Get” as well as “Set” commands. (i.e. Supported Commands fields set to 0x00)

Supported CC (N * 8 bits)

This field is used to advertise the list of Command Classes that can be scheduled by the node.

CC:0053.01.02.11.01A The length of this field MUST be according to the Number of supported CC field value.

CC:0053.01.02.11.01B A sending node MUST support every Command Class advertised in this field.

CC:0053.01.02.12.002 A sending node SHOULD NOT advertise the Supervision Command Class in this field.

CC:0053.01.02.11.01C A sending node MUST NOT advertise the following encapsulation Command Classes:

- S0/S2 Command Class
- Transport Service Command Class
- Multi Channel Command Class
- Multi Command Command Class

Supported Command (N * 2 bits)

CC:0053.01.02.11.01D This field MUST advertise the supported commands for the actual command class entry. This field MUST comply with Table 107.

Table 107. Supported Command

Value	Description	Version
0x00	Both Set and Get Commands are supported	1
0x01	Only Set Commands are supported	1
0x02	Only Get Commands are supported	1
0x03	Reserved	N/A

CC:0053.01.02.11.01E Certain command classes commands do not contain the string “Set” or “Get” in their name. Nodes MUST consider all commands mandating to return a response to be of type “Get” and all other commands to be of type “Set”.

CC:0053.01.02.11.01F The length of this field MUST be according to the Number of supported CC field value.

Override Support (1 bit)

This field is used to advertise support for the Override Schedule (Schedule ID = 0xFF).

CC:0053.01.02.11.020 This bit MUST be set to 1 if the node supports the Override Schedule.
CC:0053.01.02.11.021 This bit MUST be set to 0 if the node does not support the Override Schedule.

Supported Override Types (7 bits)

This field is used to advertise the supported Override Schedule duration types.

CC:0053.01.02.11.022 A receiving node MUST ignore this bit mask if the Override Schedule is not supported by the sending node.

CC:0053.01.02.11.023 This field MUST be treated as a bitmask and MUST be encoded according to Table 108.

Table 108. Schedule Supported Report::Supported Override Types encoding

	Bit	Name	Description	Version
CC:0053.01.02.11.024	0	Advance	The override schedule MUST run until the start of the next regular schedule.	1
CC:0053.01.02.11.025	1	Run forever	The override schedule MUST run until the schedule is removed	1
	2..6	Reserved	Reserved	N/A

4.86.5 Schedule Set Command

This command is used to create a new schedule or modify an existing schedule.

CC:0053.01.03.11.001 This command MUST enable the schedule if a new schedule is created.

CC:0053.01.03.11.002 This command MUST NOT change the enabled/disabled state of existing schedules.

CC:0053.01.03.11.003 If the receiving node supports S0 or S2 Command Class, it MUST NOT process the schedule creation if any of the scheduled command classes is not supported at the security level of the received Schedule Set Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCHEDULE							
Command = SCHEDULE_SET							
Schedule ID							
Reserved							
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 1							
Cmd Byte 1 - 1							
…							
Cmd Byte 1 - N							
…							
Cmd Length P							
Cmd Byte P - 1							
…							
Cmd Byte P - N							

Reserved

CC:0053.01.03.11.004 This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Schedule ID (8 bits)

This field is used to indicate which Schedule ID is being set.

CC:0053.01.03.11.005 Values in the range 1..[Number of supported Schedule] MUST indicate regular Schedule IDs
CC:0053.01.03.11.006 The value 0xFE MUST indicate the Fall Back Schedule.
CC:0053.01.03.11.007 The value 0xFF MUST indicate the Override Schedule.

CC:0053.01.03.11.008 A receiving node MUST ignore this command if a non-supported Schedule ID is specified.

If this field is set to 0xFF:

- CC:0053.01.03.12.001 The sending node SHOULD set the start time fields to the Start Now pattern.
- CC:0053.01.03.11.009 The receiving node MUST respond immediately to an Override Schedule .

If this field is set to 0xFE:

- CC:0053.01.03.12.002 The sending node SHOULD set the start time fields and duration fields to 0x00.
- CC:0053.01.03.11.00A The receiving node MUST ignore the start time fields and duration fields

4.86.5.1 Start time fields

The schedule start time is defined by the following fields:

- Start Year
- Start Month
- Start Day of Month
- Start Weekday
- Start Hour
- Start Minute

CC:0053.01.03.11.00B A receiving node MUST support the start time fields combinations indicated in sections 4.86.4.1, 4.86.4.2 4.86.4.3 and 4.86.4.4 if the corresponding start time option is supported.

CC:0053.01.03.11.00C If none of the start time fields are specified, the schedule MUST start immediately if the start now
CC:0053.01.03.11.00D option is supported. In this case, the schedule MUST NOT be activated again at a later time unless receiving another Schedule Set Command.

Start Year (8 bits)

CC:0053.01.03.11.00E This field is used to set the year for which the schedule is to start. This field MUST be in the range 0..99 or set to 0xFF.

CC:0053.01.03.11.00F Values in the range 0..99 MUST indicate the actual year last 2 digits. A node MUST accept a value lower than the current year to allow schedules starting in the next century.

CC:0053.01.03.11.010 The value 0xFF MUST be used to indicate that the start year is unspecified.

CC:0053.01.03.11.011 This field MUST be ignored and treated as unspecified by a receiving node advertising no support for the Calendar time start time option in the Schedule Supported Report Command.

Start Month (4 bits)

CC:0053.01.03.11.012 This field is used to set the month for which the schedule is to start. This field MUST be in the range 0..12.

CC:0053.01.03.11.013 Values in the range 1..12 MUST indicate the actual start month.

CC:0053.01.03.11.014 The value 0x00 MUST be used to indicate that the start month is unspecified.

CC:0053.01.03.11.015 This field MUST be ignored and treated as unspecified by a receiving node advertising no support for the Calendar time start time option in the Schedule Supported Report Command.

Start Day of Month (5 bits)

CC:0053.01.03.11.016 This field is used to set the day of month for which the schedule is to start. This field MUST be in the range 0..31.

CC:0053.01.03.11.017
CC:0053.01.03.13.001 Values in the range 1..31 MUST indicate the actual start day of the month. If the specified value does not exist in the actual Year/Month tuple (28/29/30 day month), a receiving node MAY ignore this command.

CC:0053.01.03.11.018 The value 0x00 MUST be used to indicate that the day of the month is unspecified.

CC:0053.01.03.11.019 This field MUST be ignored and treated as unspecified by a receiving node advertising no support for the Calendar time start time option in the Schedule Supported Report Command.

Start Weekday (7 bits)

CC:0053.01.03.11.01A This field is used to set one or more weekdays for which the schedule is to start. This field MUST be treated a as bitmask and MUST be encoded according to Table 109.

Table 109. Weekday bitmask encoding

Bit	Description	Version
0	Monday	1
1	Tuesday	1
2	Wednesday	1
3	Thursday	1
4	Friday	1
5	Saturday	1
6	Sunday	1

CC:0053.01.03.11.01B The value 1 MUST indicate that the specified Schedule ID MUST start on the corresponding day.

CC:0053.01.03.11.01C The value 0 MUST indicate that the specified Schedule ID MUST NOT start on the corresponding day.

CC:0053.01.03.11.01D If any of the Start Year, Start Month or Start Day of Month field is specified, this field MUST be set to 0x00 by a sending node and MUST be ignored by a receiving node.

CC:0053.01.03.11.01E This field MUST be ignored and treated as unspecified by a receiving node advertising no support for the Weekday start time option in the Schedule Supported Report Command.

Start Hour (5 bits)

CC:0053.01.03.11.01F This field is used to set the hour for which the schedule is to start. This field MUST be in the range 0..23 or set to 0x1F.

CC:0053.01.03.11.020 Values in the range 0..23 MUST indicate the start hour of the day.

CC:0053.01.03.11.021 The value 0x1F (31) MUST be used to indicate that the start hour is unspecified.

CC:0053.01.03.11.022 This field MUST be ignored and treated as unspecified by a receiving node advertising no support for the hour and minute start time option in the Schedule Supported Report Command.

Start Minute (6 bits)

CC:0053.01.03.11.023 This field is used to set the minute for which the schedule is to start. This field MUST be in the range 0..59 or set to 0x3F.

CC:0053.01.03.11.024 Values in the range 0..59 MUST indicate the starting minute of the schedule.

CC:0053.01.03.11.025 The value 0x3F (63) MUST be used to indicate that the start minute is unspecified.

CC:0053.01.03.11.026 This field MUST be ignored and treated as unspecified by a receiving node advertising no support for the hour and minute start time option in the Schedule Supported Report Command.

4.86.5.2 Duration fields

The schedule duration is defined by the following fields:

- Duration Type
- Duration

Duration Type (3 bits)

CC:0053.01.03.11.027 This field is used to indicate how to interpret the duration field. This field MUST be encoded according to Table 110.

Table 110. Duration Type encoding

Value	Description	Version
0x00	The duration field is expressed in Minutes	1
0x01	The duration field is expressed in Hours	1
0x02	The duration field is expressed in Days	1
0x03	Override: The duration field is indicating the Override Type. This value MUST NOT be used if the Schedule ID field is not set to 0xFF	1

CC:0053.01.03.11.028

CC:0053.01.03.11.029 All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

Duration (16 bits)

CC:0053.01.03.11.02A This field is used to specify the duration of the schedule being set. The Duration Type field MUST be inspected before interpreting this field.

CC:0053.01.03.11.02B If the Duration Type field is not set to Override (0x03), this field MUST indicate the duration of the actual schedule. The Duration Type field indicates the unit of this field.

CC:0053.01.03.11.02C If the Duration Type field is set to Override (0x03), the LSB of this field MUST be encoded according to Table 111 and the MSB of this field MUST be set to 0x00.

Table 111. Override Types duration

Name	Duration value (LSB)	Description	Version
Advance	0x01	The override schedule MUST run until the start of the next regular schedule.	1
Run forever	0x02	The override schedule MUST run until it is removed	1
Reserved	0x03..0x07	Reserved	N/A

CC:0053.01.03.11.02D

CC:0053.01.03.11.02E

4.86.5.3 Command and other fields

Reports to Follow (8 bits)

CC:0053.01.03.11.02F This field MUST be used if multiple Schedule Set commands are used to define a single schedule.

CC:0053.01.03.11.030 The value MUST indicate the remaining number of Schedule Set Commands. The header bytes (Schedule ID, Start time, Duration, etc.) MUST be the same for all Schedule Set Commands.

Number of Cmd to Follow (8 bits)

This field is used to advertise the number of command blocks (Cmd Length/Cmd Byte fields units) included within the actual Schedule Set Command (represented by P in the command structure).

CC:0053.01.03.11.031 Each command block MUST comprise the Cmd Length and Cmd Byte fields.

CC:0053.01.03.11.032
CC:0053.01.03.11.033 If the list of scheduled commands and their payload is longer than the maximum available Z-Wave MAC frame size, multiple Schedule Set commands MUST be used to send the complete list of scheduled commands. In this case, this field MUST advertise how many commands are included in the actual/current Schedule Set Command.

Cmd Length (8 bits)

This field is used to indicate the length in byte of the corresponding Cmd Byte field.

Cmd Byte (N bytes)

This field is used to set the commands to be executed during the actual Schedule.

CC:0053.01.03.11.034 This field MUST carry a complete command including Command Class identifier, Command identifier and the required parameter bytes.

CC:0053.01.03.11.035 The length of each Cmd Byte entry MUST be according to the corresponding Cmd Length field entry.

CC:0053.01.03.11.036 The number of Cmd Byte entries MUST be according to the Number of Cmd to Follow field.

CC:0053.01.03.11.037 If a schedule includes commands mandating to return a response, responses MUST be returned to the node that has set the schedule.

CC:0053.01.03.12.003 A receiving node SHOULD ignore the entire Schedule Set Command if this field carries a command not advertised as supported for scheduling in the Schedule Supported Report Command. In any case, the receiving node MUST NOT accept commands non-supported for scheduling as part of the Schedule.

CC:0053.01.03.11.038

CC:0053.01.03.12.004 A controlling node SHOULD avoid creating conflicting schedules and a supporting node SHOULD discard a new conflicting schedule, even if the conflict is with a temporarily disabled schedule.

4.86.6 Schedule Get Command

This command is used to request the configuration for a specific schedule ID.

CC:0053.01.04.11.001 The Schedule Report Command MUST be returned in response to this command.

CC:0053.01.04.11.002 This command MUST NOT be issued via multicast addressing.

CC:0053.01.04.11.003 A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCHEDULE							
Command = SCHEDULE_GET							
Schedule ID							

Schedule ID (8 bits)

CC:0053.01.04.11.004 This field MUST carry the requested Schedule ID.

4.86.7 Schedule Report Command

This command is used to report the configuration for a specific schedule.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCHEDULE							
Command = SCHEDULE_REPORT							
Schedule ID							
Reserved							
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 1							
Cmd Byte 1 - 1							
…							
Cmd Byte 1 – N							
…							
Cmd Length P							
Cmd Byte P - 1							
…							
Cmd Byte P - N							

Fields not described below are described in the Schedule Set Command; refer to 4.86.5.

Schedule ID (8 bits)

This field is used to indicate the advertised Schedule ID.

CC:0053.01.05.11.001 If this command is advertising an unused or unsupported Schedule ID, all parameters until Number of Cmd to Follow MUST be set to 0x00. The Cmd Length and Cmd Byte fields MUST be omitted.

Active_ID (4 bits)

CC:0053.01.05.11.002 This field is used to advertise the status for the requested Schedule ID. This field MUST be encoded according to Table 113.

Duration (16 bits)

This field is used to advertise either the configured duration for a schedule or how much time is left before the schedule becomes inactive.

CC:0053.01.05.11.003 The Duration Type, Start Time and Schedule ID fields MUST be inspected before interpreting this field. This field MUST advertise a value according to Table 112.

Table 112. Duration field usage

Duration Type	Schedule ID	Start time	Advertised duration
Time (0x00..0x02)	Regular	Specified start time	Configured duration
Time (0x00..0x02)	Regular or Override	Start now	Remaining time
Override (0x03)	Override (0xFF)	Start now	Configured override type (Table 111)

4.86.8 Schedule Remove Command

This command is used to request the removal of one or all Schedules in a device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCHEDULE							
Command = SCHEDULE_REMOVE							
Schedule ID							

Schedule ID (8 bits)

This field is used to indicate the schedule that is to be removed/erased.

CC:0053.01.06.11.001 If this field is set to 0x00, all configured schedules MUST be removed.
CC:0053.01.06.11.002 If this field is set to 0xFE, the Fall Back Schedule MUST be removed (if defined).
CC:0053.01.06.11.003 If this field is set to 0xFF, the Override Schedule MUST be removed (if defined).
CC:0053.01.06.11.004 If this field is in the range 0x01..[Number of supported Schedules], the receiving node MUST remove the corresponding schedule.

4.86.9 Schedule State Set Command

This command is used to enable or disable a schedule.

CC:0053.01.07.11.001 This command MUST be ignored by a node advertising no support for Enabling/Disabling schedules in the Schedule Supported Report Command.

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)

This field is used to indicate the schedule that is to be enabled or disabled.

CC:0053.01.07.11.002 The value 0x00 MUST cause the receiving node to change the state for all schedules, except for the Override Schedule.

CC:0053.01.07.11.003 Values in the range 0x01..[number of supported schedules] and 0xFE MUST indicate the actual schedule ID which state is to be updated.

CC:0053.01.07.11.004 The value 0xFF MUST be ignored by a receiving node. Override schedules cannot be enabled/disabled and may only be removed using the Schedule Remove Command.

Schedule state (8 bits)

This field is used to indicate the new state of one or more schedules.

CC:0053.01.07.11.005 The value 0xFF MUST indicate that the actual schedule MUST be enabled.
CC:0053.01.07.11.006 The value 0x00 MUST indicate that the actual schedule MUST be disabled.

CC:0053.01.07.11.007 Disabling a schedule MUST NOT cause the schedule settings to be permanently removed. Schedules may be permanently removed via the Schedule Remove Command.

CC:0053.01.07.11.008 If a schedule is enabled, the receiving node MUST assess if the actual schedule should have been active and activate it accordingly.

CC:0053.01.07.11.009 If a schedule is disabled while being active, the receiving node MUST treat the disabling operation as if the schedule ended.

4.86.10 Schedule State Get Command

This command is used to request the status of all schedules supported by a node.

CC:0053.01.08.11.001 The Schedule State Report Command MUST be returned in response to this command.

CC:0053.01.08.11.002 This command MUST NOT be issued via multicast addressing.

CC:0053.01.08.11.003 A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCHEDULE							
Command = SCHEDULE_STATE_GET							

4.86.11 Schedule State Report Command

This command is used to advertise the status of all schedules supported by a device.

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 N-1			

.

Number of Supported Schedule ID (8 bits)

CC:0053.01.09.11.001 This field is used to advertise the number of supported regular schedule IDs. This field's value MUST be the same as the value advertised in the Schedule Supported Report Command.

Override (1 bit)

CC:0053.01.09.11.002 This field MUST be used to advertise the status of the Override Schedule (ID = 0xFF).

CC:0053.01.09.11.003 If the Override Schedule is active, this bit MUST be set to 1.

CC:0053.01.09.11.004 If the Override Schedule is inactive, this bit MUST be set to 0.

Reports to Follow (7 bits)

The length of the Active_ID field may be larger than the maximum available Z-Wave MAC frame size.

CC:0053.01.09.11.005 This field MUST be used to advertise the number of commands following this command in order to advertise the status of all Schedules IDs.

CC:0053.01.09.12.001 A controlling node SHOULD use this value to detect missing reports.

Active_ID (N * 4 bits)

This field is used to advertise the status for each supported schedule ID. Four bits units are used for each Schedule ID to indicate the status. It means that:

- Bits 0 to 3 in Active_ID byte 1 represent Schedule ID = 1
- Bits 4 to 7 in Active_ID byte 1 represent Schedule ID = 2
- Bits 0 to 3 in Active_ID byte 2 represent Schedule ID = 3
- …

CC:0053.01.09.11.006 The size of this field MUST be the smallest number of bytes needed to advertise the number of supported Schedule IDs advertised in this command.

CC:0053.01.09.11.007 If several Reports are returned (using the Reports to follow field), the first Report advertises Schedule ID 1 to Schedule ID N and the subsequent report MUST be interpreted as representing Schedule ID starting from N+1.

CC:0053.01.09.11.008 Each 4 bits unit MUST be encoded according to Table 113.

Table 113. Active_ID encoding

Hex	Description	Detailed description	Version
0x00	Not used	The Schedule ID is not used. (not set/configured) or unsupported	1
0x01	[DEPRECATED] Override + Not used	[DEPRECATED] A sending node SHOULD NOT use this status. It is RECOMMENDED to use 0x00 instead. A receiving node MUST interpret this status as Active_ID = 0x00 (and the override field set to 1 for the Schedule State Report Command).	1
0x02	Not Active	The Schedule ID is used, enabled and currently not active.	1
0x03	Active	The Schedule ID is used, enabled and currently active.	1
0x04	Disabled	The Schedule ID is used and disabled.	1
0x05	Override + Active	The Schedule ID is used, enabled and should currently be active but it is suspended by the Override Schedule The override field MUST be set to 1 when using this status in the Schedule State Report Command.	1
0x06	[DEPRECATED] Override + Not Active	[DEPRECATED] A sending node SHOULD NOT use this status. It is RECOMMENDED to use 0x02 instead. A receiving node MUST interpret this status as Active_ID = 0x02 (and the override field set to 1 for the Schedule State Report Command)	1
0x07	[DEPRECATED] Override + Disabled	[DEPRECATED] A sending node SHOULD NOT use this status. It is RECOMMENDED to use 0x04 instead. A receiving node MUST interpret this status as Active_ID = 0x04 (and the override field set to 1 for the Schedule State Report Command)	1

CC:0053.01.09.12.002
CC:0053.01.09.11.009
CC:0053.01.09.11.00A
CC:0053.01.09.12.003
CC:0053.01.09.11.00B
CC:0053.01.09.12.004
CC:0053.01.09.11.00C

CC:0053.01.09.11.00D All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

4.87 Schedule Command Class, version 2

The Schedule Command Class version 2 allows scheduling the execution of commands for a given duration in a supporting node. It is a generic command class that may be used to schedule commands from any other Command Class.

4.87.1 Compatibility considerations

The Schedule Command Class, version 2 is backwards compatible with version 1.

CC:0053.02.00.21.001 A node supporting the Schedule Command Class, version 2 MUST also support version 1.

Schedule Command Class, version 2 introduces the following:

- Schedule ID Blocks
- Clarification on using Schedule Command Class with Security

All commands or fields not described in this version remain unchanged from version 1.

4.87.1.1 Schedule ID Blocks

In Schedule Command Class version 1, all Schedule IDs have to support the same Start Time Options, Override Types and supported commands.

The Schedule ID Blocks allows a node to implement several Schedule ID pools sharing the same scheduling functionalities.

CC:0053.02.00.21.002 Each Schedule ID Block MUST have its own range of Schedule ID's. The ID for each Block MUST be in the range 1..[Number of Supported Schedule IDs].

CC:0053.02.00.21.003 Schedule ID Block = 1 is the default Block. A version 1 node MUST be assumed to be using Schedule ID block 1.

4.87.1.2 Schedule Command Class with Security

The Schedule Command Class may be implemented by secure devices. Depending on the type of device, a given Command Class may be supported only via secure communication or via secure as well as unsecure communication.

The following command class categories may be considered:

- **Always unsecure**
  Examples of command classes which are always supported via unsecure communication – as well as via secure communication if the device is securely included (e.g Z-Wave Plus Info Command Class).
- **Migrate to secure**
  Examples of command classes which are supported via unsecure communication if the device is not securely included – but only via secure communication if the device is securely included. (e.g. Multilevel Switch or Association Command Class)
- **Secure only:**
  Examples of command classes which are supported only via secure communication. (e.g. Door Lock Command Class)

CC:0053.02.00.21.004 To reflect the above mentioned dynamic support scenarios, a device MUST advertise the supported Command Classes for scheduling in a way that matches the current secure/non-secure supported Command Class lists found in the NIF and S0/S2 Commands Supported Report Commands.

Clarification is added to the Schedule Supported Report Command and Schedule Set Command for proper reporting and configuration of unsecure and secure command classes scheduling.

CC:0053.02.00.22.001 It is RECOMMENDED to support the Schedule Command Class only at the highest security level in order to avoid dynamic listing.

4.87.2 Schedule Supported Get Command

This command is used to query the properties of a node.

CC:0053.02.01.11.001 The Schedule Supported Report Command MUST be returned in response to this command.

CC:0053.02.01.11.002 If the receiving node supports S0 or S2 Command Class and this command is issued via non-secure communication, the receiving node MUST advertise Command Classes that can be scheduled but also supported at the received security level.

CC:0053.02.01.11.003 This command MUST NOT be issued via multicast addressing.

CC:0053.02.01.11.004 A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCHEDULE							
Command = SCHEDULE_SUPPORTED_GET							
Schedule ID Block							

Schedule ID Block (1 byte)

This field is used to request a particular Schedule ID Block.

CC:0053.02.01.11.005 If this field is set to 0x00, the receiving node MUST return the default Schedule ID Block (ID = 1).

CC:0053.02.01.11.006 A node receiving a version 1 Schedule Supported Get Command (without the Schedule ID Block field) MUST return a response for the default Schedule ID Block (ID = 1).

CC:0053.02.01.12.001 A node receiving a non-supported Schedule ID Block SHOULD return a report for the default Schedule ID Block.

4.87.3 Schedule Supported Report Command

This command is used to advertise the scheduling functionalities supported by a node for a given Schedule ID Block.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCHEDULE							
Command = SCHEDULE_SUPPORTED_REPORT							
Number of Supported Schedule IDs							
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
Override Support	Supported Override Types						
Schedule ID Block							
Number of Supported Schedule Blocks							

Fields not described below remain unchanged from version 1.

All fields contained in this command advertise the scheduling supported functionalities for the actual Schedule ID Block.

Schedule ID Block (8 bits)

This field is used to indicate the actual Schedule ID Block for which the supported scheduling functionalities are advertised.

CC:0053.02.02.12.001 The advertised supported functionalities SHOULD be different for every Schedule ID Block.

Number of Schedule ID Blocks (8 bits)

CC:0053.02.02.11.001 This field MUST advertise the total number of Schedule ID blocks supported by the node.

CC:0053.02.02.11.002 This field MUST be in the range 1..255.

CC:0053.02.02.11.003 The implemented Schedule ID Blocks MUST be in the range 1..[Number of Supported Schedule ID Blocks]. i.e. a node supporting 10 Schedules ID Blocks MUST accept Schedule ID Blocks values in the range 1..10.

4.87.4 Schedule Set Command

This command is used to create a new schedule or modify an existing schedule.

CC:0053.02.03.11.001 This command MUST enable the schedule if a new schedule is created.

CC:0053.02.03.11.002 This command MUST NOT change the enabled/disabled state of existing schedules

CC:0053.02.03.11.003 If the receiving node supports S0 or S2 Command Class, it MUST NOT process the schedule creation if any of the scheduled command classes is not supported at the security level of the received Schedule Set Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCHEDULE							
Command = SCHEDULE_SET							
Schedule ID							
Schedule ID Block							
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 1							
Cmd 1 Byte 1 - 1							
…							
Cmd 1 Byte 1 - N							
…							
Cmd Length P							
Cmd Byte P - 1							
…							
Cmd Byte P - N							

Fields not described below remain unchanged from version 1.

Schedule ID Block (8 bits)

This field is used to specify the Schedule ID Block in which the specified Schedule ID is being set.

CC:0053.02.03.11.004 A node receiving a version 1 Schedule Set Command (with the Schedule ID Block field set to 0x00) MUST assume the default Schedule ID Block (ID = 0x01).

4.87.5 Schedule Get Command

This command is used to request the configuration for a specific schedule ID.

CC:0053.02.04.11.001 The Schedule Report Command MUST be returned in response to this command.

CC:0053.02.04.11.002 This command MUST NOT be issued via multicast addressing.

CC:0053.02.04.11.003 A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCHEDULE							
Command = SCHEDULE_GET							
Schedule ID							
Schedule ID Block							

Fields not described below remain unchanged from version 1.

Schedule ID Block (8 bits)

This field is used to specify the requested Schedule ID Block.

CC:0053.02.04.11.004 A node receiving a version 1 Schedule Get Command (without the Schedule ID Block field) MUST return a response for the default Schedule ID Block (ID = 1).

4.87.6 Schedule Report Command

This command is used to advertise the configuration for a specific schedule.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCHEDULE							
Command = SCHEDULE_REPORT							
Schedule ID							
Schedule ID Block							
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 1							
Cmd Byte 1 - 1							
…							
Cmd Byte 1 – N							
…							
Cmd Length P							
Cmd Byte P - 1							
…							
Cmd Byte P - N							

Fields not described below remain unchanged from version 1.

Schedule ID Block (8 bits)

This field is used to specify the advertised Schedule ID Block.

4.87.7 Schedule Remove Command

This command is used to request the removal of one or all Schedules in a device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCHEDULE							
Command = SCHEDULE_REMOVE							
Schedule ID							
Schedule ID Block							

Schedule ID (8 bits)

This field is used to indicate the schedule that is to be removed.

CC:0053.02.06.11.001 If this field is set to 0x00, all schedules within the Schedule ID Block MUST be removed.

CC:0053.02.06.11.002 If this field is set to 0xFE, the Fall Back Schedule MUST be removed (if defined for the Schedule ID Block).

CC:0053.02.06.11.003 If this field is set to 0xFF, the Override Schedule MUST be removed (if defined for the Schedule ID Block).

CC:0053.02.06.11.004 If this field is in the range 0x01..[Number of supported Schedules], the receiving node MUST remove the corresponding schedule in the Schedule ID Block.

Schedule ID Block (8 bits)

This field is used to indicate the Schedule ID Block in which a schedule is being removed.

CC:0053.02.06.11.005 A node receiving a version 1 Schedule Set Command (without the Schedule ID Block field) MUST assume the default Schedule ID Block (ID = 0x01).

CC:0053.02.06.11.006 A node receiving this command with **Schedule ID = 0x00** and **Schedule ID Block = 0x00** MUST remove all Schedule IDs in all Schedule ID Blocks

CC:0053.02.06.11.007 A node receiving this command with **Schedule ID = 0x00** and **Schedule ID Block ≠ 0x00** MUST remove all Schedule IDs from the specified Schedule ID Block.

CC:0053.02.06.11.008 A node receiving this command with **Schedule ID ≠ 0x00** and **Schedule ID Block = 0x00** MUST ignore the command

4.87.8 Schedule State Set Command

This command is used to enable or disable one or more schedules.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCHEDULE							
Command = SCHEDULE_STATE_SET							
Schedule ID							
Schedule State							
Schedule ID Block							

Fields not described below remain unchanged from version 1.

Schedule ID Block (8 bits)

This field is used to specify the Schedule ID Block in which one or more schedule are being enabled or disabled.

CC:0053.02.07.11.001 This command MUST be ignored by a node advertising no support for Enabling/Disabling schedules in the Schedule Supported Report Command for the given Schedule ID Block.

CC:0053.02.07.11.002 A node receiving a version 1 Schedule Set Command (without the Schedule ID Block field) MUST assume the default Schedule ID Block (ID = 0x01).

CC:0053.02.07.11.003 A node receiving this command with **Schedule ID = 0x00** and **Schedule ID Block = 0x00** MUST enable/disable all Schedule IDs from all Schedule ID Blocks.

CC:0053.02.07.11.004 A node receiving this command with **Schedule ID = 0x00** and **Schedule ID Block ≠ 0x00** MUST enable/disable all Schedule IDs from the specified Schedule ID Block.

CC:0053.02.07.11.005 A node receiving this command with **Schedule ID ≠ 0x00** and **Schedule ID Block = 0x00** MUST ignore the command.

4.87.9 Schedule State Get Command

This command is used to request the status of all schedules within a Schedule ID Block supported by a node.

CC:0053.02.08.11.001 The Schedule State Report Command MUST be returned in response to this command.

CC:0053.02.08.11.002 This command MUST NOT be issued via multicast addressing.

CC:0053.02.08.11.003 A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCHEDULE							
Command = SCHEDULE_STATE_GET							
Schedule ID Block							

Schedule ID Block (8 bits)

This field is used to request a particular Schedule ID Block.

CC:0053.02.08.11.004 A node receiving a version 1 Schedule Supported Get Command (without the Schedule ID Block field) MUST return a response for the default Schedule ID Block (ID = 1).

4.87.10 Schedule State Report Command

This command is used to advertise the status of all schedules supported by a device in a given Schedule ID block.

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 N-1			
Schedule ID Block							

Fields not described below remain unchanged from version 1.

Number of Supported Schedule ID (8 bits)

CC:0053.02.09.11.001 This field is used to advertise the number of supported regular schedule IDs for the actual Schedule ID Block. This field's value MUST be the same as the value advertised in the Schedule Supported Report Command.

Schedule ID Block (8 bits)

This field is used to specify the Schedule ID Block in which all schedule states are advertised.

CC:0053.02.09.11.002 If several Reports are returned (using the Reports to Follow field), this field MUST be present in every Report.

4.88 Schedule Command Class, version 3

The Schedule Command Class, version 3 allows scheduling the execution of commands for a given duration in a supporting node. It is a generic command class that may be used to schedule commands of any other Command Class.

4.88.1 Terminology

The Schedule Command Class version 3 introduces new terminology, complementing the terminology introduced in The Schedule Command Class version 1:

Regular schedules may be triggered repeatedly by two mechanisms.
From version 1, a Repeating Schedule can be created by setting parts of the start time to unspecified values, e.g. if month is unspecified, the schedule can start the first day of each month.
From version 3, Recurrence settings may specify additional periodical triggers, e.g. every second day measured from the most recent trigger of the regular schedule.

4.88.2 Compatibility considerations

The Schedule Command Class, version 3 is backwards compatible with version 2.

CC:0053.03.00.21.001 A node supporting the Schedule Command Class, version 3 MUST also support version 2.

The Schedule Command Class version 3 introduces the “Recurring Mode” start mode and “Time from now” start time option.

The “Time from now” start time option allows to set a schedule to start a after a given time from the reception of the command.

The “Recurring Mode” start mode allows to set a schedule to restart repeatedly at fixed intervals since the last activation, e.g. every second day or every 6 months since the last activation.

The Active_ID advertised in the Schedule Report Command of versions 1 and 2 is now overloaded with the Recurrence Offset field. A new AID_RO_CTL flag in the Schedule Get Command controls which value is actually returned.

All commands or fields not described in this version remain unchanged from version 2.

4.88.3 Schedule Supported Report Command

This command is used to advertise the scheduling functionalities supported by a node.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCHEDULE							
Command = SCHEDULE_SUPPORTED_REPORT							
Number of Supported Schedule IDs							
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	
Override Support	Supported Override Types						
Schedule ID Block							
Number of Supported Schedule Blocks							

Fields not described below remain unchanged from version 2.

Start Time Support (6 bits)

This field is used to advertise the start time options supported by the sending node.

CC:0053.03.02.11.001 Regular schedules MUST support the advertised start time options.
CC:0053.03.02.12.001 The Override Schedule SHOULD support the advertised start time options.
CC:0053.03.02.11.002 The Override Schedule MUST NOT support the Recurring Mode.
CC:0053.03.02.11.003 This field MUST be treated as a bitmask and MUST be encoded according to Table 114

Table 114. Start Time Support encoding

Category	Bit	Indicates support for	Version
Start Time option	0	Start now. Refer to 4.86.4.1	1
	1	Start Hour and Minute. Refer to 4.86.4.2	1
	2	Calendar time. Refer to 4.86.4.3	1
	3	Weekdays. Refer to 4.86.4.4	1
	4	Time from now. Refer to 4.88.3.1	3
Start Time mode	5	Recurring Mode. Refer to 4.88.3.2	3

Each bit indicates the support for a given Start Time option or mode.

CC:0053.03.02.11.004 The value 1 MUST indicate that the node supports the corresponding start time option/mode.
CC:0053.03.02.11.005 The value 0 MUST indicate that the node does not support the corresponding start time option/mode.

While support is advertised as a bitmap, the actual functionality is triggered by different combinations of Schedule Set start time fields. The following subsections outline the mandatory supported combinations when advertising support for the corresponding option.

4.88.3.1 Start Time option: Start from now

The *start now* option is used to make a configured schedule start after the indicated time has elapsed from the reception of the Schedule Set Command.

CC:0053.03.02.11.006 The *Start from now* option MUST cause a schedule to be activated at the specified relative time measured from the reception of the *Schedule Set* Command and run for the specified duration.

CC:0053.03.02.11.007 A node supporting the *Time from now* option MUST support the creation of a schedule with the following start time fields' combination:

Relative = '1' and
YYMMDD = 0xFF, 0x00, **Days**, Weekdays = 0x00, HH:MM = **Hours:Minutes**

CC:0053.03.02.11.008 A receiving node MUST ignore all combinations which do not comply with the one above if the Relative flag is set.

CC:0053.03.02.11.009 A node supporting this option is NOT REQUIRED to support the Start hour and Minute option or Calendar time option even though it MUST be able to read the start day, start hour and start minutes fields when using this option.

4.88.3.2 Start Time mode: Recurring Mode

This mode is used to trigger a schedule repeatedly at fixed intervals. When using this mode, a schedule will start at the specified start time and will restart after a given time has elapsed since the last activation.

The recurring mode is configured with the Recurrence Offset and Recurrence Mode fields of the Schedule Set Command.

CC:0053.03.02.11.00A A node supporting the Recurring Mode MUST support at least one start time option.

4.88.4 Schedule Set Command

This command is used to create a new schedule or modify an existing schedule

CC:0053.03.03.11.001 This command MUST enable the schedule if a new schedule is created.

CC:0053.03.03.11.002 This command MUST NOT change the enabled/disabled state of existing schedules.

CC:0053.03.03.11.003 If the receiving node supports S0 or S2 Command Class, it MUST NOT process the schedule creation if any of the scheduled command classes is not supported at the security level of the received Schedule Set Command.

<table>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="8">Command Class = COMMAND_CLASS_SCHEDULE</td></tr>
<tr><td colspan="8">Command = SCHEDULE_SET</td></tr>
<tr><td colspan="8">Schedule ID</td></tr>
<tr><td colspan="8">Schedule ID Block</td></tr>
<tr><td colspan="8">Start Year</td></tr>
<tr><td colspan="4">Recurrence Offset</td><td colspan="4">Start Month</td></tr>
<tr><td>Res.</td><td colspan="2">Recurrence Mode</td><td colspan="5">Start Day of Month</td></tr>
<tr><td>Res.</td><td colspan="7">Start Weekday</td></tr>
<tr><td colspan="3">Duration Type</td><td colspan="5">Start Hour</td></tr>
<tr><td>Res.</td><td>Relative</td><td colspan="6">Start Minute</td></tr>
<tr><td colspan="8">Duration Byte 1 MSB</td></tr>
<tr><td colspan="8">Duration Byte 2 LSB</td></tr>
<tr><td colspan="8">Reports to Follow</td></tr>
<tr><td colspan="8">Number of Cmd to Follow</td></tr>
<tr><td colspan="8">Cmd Length 1</td></tr>
<tr><td colspan="8">Cmd Byte 1 - 1</td></tr>
<tr><td colspan="8">…</td></tr>
<tr><td colspan="8">Cmd Byte 1 - N</td></tr>
<tr><td colspan="8">…</td></tr>
<tr><td colspan="8">Cmd Length P</td></tr>
<tr><td colspan="8">Cmd Byte P - 1</td></tr>
<tr><td colspan="8">…</td></tr>
<tr><td colspan="8">Cmd Byte P - N</td></tr>
</table>

Fields not described below remain unchanged from version 2.

Reserved

CC:0053.03.03.11.004 This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

4.88.4.1 Recurrence fields

The recurrence settings of a schedule are set by the following fields:

- Recurrence Mode
- Recurrence Offset

CC:0053.03.03.11.005 These fields MUST be ignored by a node advertising no support for the Recurring Mode start time mode in the Schedule Supported Report Command.

Recurrence mode (2 bits)

This field is used to specify the unit of the Recurrence Offset field.

CC:0053.03.03.11.006 This field MUST comply with Table 115.

Table 115. Recurrence Mode encoding

Value	Description	Version
0x00	The recurrence Offset is expressed in Hours	3
0x01	The recurrence Offset is expressed in Days	3
0x02	The recurrence Offset is expressed in Weeks	3

CC:0053.03.03.11.007 All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

CC:0053.03.03.11.008 This field MUST be ignored if the Recurrence Offset field is set to 0.

Recurrence Offset (4 bits)

This field is used to specify the interval at which the recurring schedule will be triggered.

CC:0053.03.03.11.009 This field MUST be encoded according to Table 116 and MUST be interpreted together with the Recurrence Mode field.

Table 116. Recurrence Offset encoding

Offset	Description
0	Recurrence Disabled
1..15	Repeat every 1..15 hour, day or week The unit (hour, day or week) is indicated by the Recurrence Mode field.

CC:0053.03.03.11.00A
CC:0053.03.03.11.00B
CC:0053.03.03.11.00C When a schedule starts due to its start time options, it MUST restart the current recurrence timer. It means that a schedule MUST start every time the start time condition is met and the recurrence offset MUST trigger the schedule to restart based on the last time it started.

CC:0053.03.03.11.00D For example, a schedule set to start on the 1st of each month with a 2 day recurrence offset MUST start on the 1st, 3rd, 5th …, 27th, 29th, 31st and will start again the next month on the 1st, 3rd, etc.

CC:0053.03.03.11.00E A receiving node MUST accept a schedule set to start in the past if recurrence is enabled. In this case, the recurrence MUST trigger the schedule to restart based on the time it should have been started last.

4.88.4.2 Start time fields

The Schedule start time is defined by the following fields:

- Start Year
- Start Month
- Start Day of Month
- Start Weekday
- Start Hour
- Start Minute
- Relative

CC:0053.03.03.11.00F A receiving node MUST support the start time fields combinations indicated in sections 4.86.4.1, 4.86.4.2 4.86.4.3, 4.86.4.4 and 4.88.3.1 if the corresponding start time option is supported.

CC:0053.03.03.11.010 If none of the start time fields are specified, the schedule MUST start immediately if the start now
CC:0053.03.03.11.011 option is supported. In this case, the schedule MUST NOT be activated again at a later time unless receiving another Schedule Set Command.

Relative (1 bit)

This field is used to indicate if the schedule start time is using the start from now start option.

CC:0053.03.03.11.012 The value 1 MUST indicate that the start time fields MUST be interpreted as the time left before the
CC:0053.03.03.11.013 schedule starts. In this case, the start time fields MUST be set and interpreted as indicated in 4.88.3.1
CC:0053.03.03.11.014 and the schedule MUST NOT be activated again at a later time unless receiving another Schedule Set Command.

CC:0053.03.03.11.015 The value 0 MUST indicate that the start time fields MUST be interpreted as an absolute start time for the schedule.

4.88.5 Schedule Get Command

This command is used to request the configuration for a specific schedule ID.

CC:0053.03.04.11.001 The Schedule Report Command MUST be returned in response to this command.

CC:0053.03.04.11.002 This command MUST NOT be issued via multicast addressing.

CC:0053.03.04.11.003 A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCHEDULE							
Command = SCHEDULE_GET							
Schedule ID							
Schedule ID Block							
AID_RO_CTL	Reserved						

Fields not described below remain unchanged from version 2.

Active_ID / Recurrence Offset control (AID_RO_CTL) (1 bit)

This field is used to request either the Active_ID or the Recurrence Offset value in the Schedule Report Command which is returned in response to this command.

CC:0053.03.04.11.004 A receiving node MUST return a Schedule Report Command advertising the Active_ID of the Schedule if this field is set to 0.

CC:0053.03.04.11.005 A receiving node MUST return a Schedule Report Command advertising the Recurrence Offset of the schedule if this field is set to 1.

4.88.6 Schedule Report Command

The Schedule Report Command is used to advertise the configuration for a specific schedule.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCHEDULE							
Command = SCHEDULE_REPORT							
Schedule ID							
Schedule ID Block							
Start Year							
Active_ID/Recurrence Offset (AID_RO)				Start Month			
AID_RO_CTL	Recurrence Mode		Start Day of Month				
Res.	Start Weekday						
Duration Type			Start Hour				
Res.	Relative	Start Minute					
Duration Byte 1							
Duration Byte 2							
Report to Follow							
Number of Cmd to Follow							
Cmd Length 1							
Cmd Byte 1 - 1							
…							
Cmd Byte 1 – N							
…							
Cmd Length P							
Cmd Byte P - 1							
…							
Cmd Byte P - N							

Fields not described below remain unchanged from version 2.

Active_ID/Recurrence Offset (AID_RO) (4 bits)

This field is used to advertise the Active_ID or the Recurrence Offset of the advertised schedule.

CC:0053.03.05.11.001 If this field carries the Active_ID value, this field MUST be encoded according to Table 113.

CC:0053.03.05.11.002 If this field carries the Recurrence Offset, this field MUST be encoded according to Table 116 and MUST be interpreted together with the Recurrence Mode field.

Active_ID / Recurrence Offset control (AID_RO_CTL) (1 bit)

This field is used to indicate whether the Active_ID or the Recurrence Offset value is advertised in the command.

CC:0053.03.05.11.003 The value 0 MUST indicate that the Active_ID/Recurrence Offset (AID_RO) field carried the Active_ID value.

CC:0053.03.05.11.004 The value 1 MUST indicate that the Active_ID/Recurrence Offset (AID_RO) field carried the Recurrence Offset value.

Recurrence mode (2 bits)

This field is used to specify the Recurrence Offset unit.

CC:0053.03.05.11.005 This field MUST be ignored if the AID_RO_CTL field is set to 0.

CC:0053.03.05.11.006 This field MUST comply with Table 115 if the AID_RO_CTL field is set to 1.

4.89 Schedule Command Class, version 4

The Schedule Command Class, version 4 allows supporting nodes to advertise the list of commands they support for scheduling.

4.89.1 Compatibility considerations

The Schedule Command Class version 4 introduces the possibility to advertise the list of command supported for scheduling and explicitly allows scheduling extended command classes. The following commands are introduced:

- Schedule Supported Commands Get Command
- Schedule Supported Commands Report Command

The following functionalities become obsoleted:

- Redundant values in the Active_ID field encoding
- The Report to Follow field.

A supporting node no longer needs to be able to parse and handle several commands in order to set or report schedules or schedules states. Supporting nodes MUST use the Transport Service Command Class, version 2 in order to transmit long Z-Wave payloads.

CC:0053.04.00.21.001 A node supporting the Schedule Command Class, version 4 MUST also support version 3. Commands and fields not mentioned in this version MUST remain unchanged from version 3.

4.89.2 Schedule Supported Report Command

This command is used to advertise the scheduling properties of a device.

<table>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="8">Command Class = COMMAND_CLASS_SCHEDULE</td></tr>
<tr><td colspan="8">Command = SCHEDULE_SUPPORTED_REPORT (0x02)</td></tr>
<tr><td colspan="8">Number of Supported Schedule IDs</td></tr>
<tr><td>Support Enable/Disable</td><td>Fall-back Support</td><td colspan="6">Start Time Support</td></tr>
<tr><td colspan="8">Number of supported CC</td></tr>
<tr><td colspan="8">Supported CC 1</td></tr>
<tr><td colspan="7">Reserved</td><td>Supported Command 1</td></tr>
<tr><td colspan="8">…</td></tr>
<tr><td colspan="8">Supported CC N</td></tr>
<tr><td colspan="7">Reserved</td><td>Supported Command N</td></tr>
</table>

Override Support	Supported Override Types
Schedule ID Block	
Number of Supported Schedule Blocks	

All fields not described below remain unchanged from version 3.

Supported Command (N * 2 bits)

CC:0053.04.02.11.001 This field MUST advertise the supported commands for the command class entry. This field MUST comply with Table 117.

Table 117. Schedule Supported Report v4::Supported Command

Value	Description	Version
0x00	All Commands within the Command Class are supported	1
0x01	Only Set Commands are supported	1
0x02	Only Get Commands are supported	1
0x03	Custom list of commands are supported.	4

CC:0053.04.02.11.002 Certain command classes commands do not contain the string “Set” or “Get” in their name. Nodes MUST consider all commands mandating to return a response to be of type “Get” and all other commands to be of type “Set”.

CC:0053.04.02.11.003 The length of this field in bytes MUST be according to the *Number of supported CC* field value.

CC:0053.04.02.11.004 A supporting node MUST NOT advertise the 0x03 value if the list of supported command can be advertised with the other values.

CC:0053.04.02.52.001 If a sending node advertises 0x03 for a given Schedule ID Block and Command Class, a controlling node SHOULD retrieve the exact list of supported commands using the Schedule Supported Commands Get Command.

4.89.3 Schedule Supported Commands Get Command

This command is used to query the list of commands that can be scheduled within for a given command class and Schedule ID Block.

CC:0053.04.0A.11.001 The Schedule Supported Commands Report Command MUST be returned in response to this command.

CC:0053.04.0A.51.001 This command MUST NOT be issued via multicast addressing.

CC:0053.04.0A.11.002 A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCHEDULE							
Command = SCHEDULE_SUPPORTED_COMMANDS_GET (0x0A)							
Schedule ID Block							

Schedule ID Block (8 bits)

This field is used to request a particular Schedule ID Block.

CC:0053.04.0A.12.001 If a non-supported Schedule ID Block is specified, a receiving node SHOULD return a response for the Schedule ID block 1.

4.89.4 Schedule Supported Commands Report Command

This command is used to advertise the list of commands that can be scheduled within a given Schedule ID Block

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCHEDULE							
Command = SCHEDULE_SUPPORTED_COMMANDS_REPORT (0x0B)							
Schedule ID Block							
Command Class List Length							
Command Class 1 (1 or 2 bytes)							
Supported Command List Length 1							
Supported Command 1 – 1							
…							
Supported Command 1 – K							
…							
Command Class N (1 or 2 bytes)							
Supported Command List Length N							
Supported Command N – 1							
…							
Supported Command N – K							

Schedule ID Block (8 bits)

This field is used to specify the requested Schedule ID Block.

Command Class List Length (8 bit)

This field is used to advertise the number of command classes contained in the command.

CC:0053.04.0B.11.001 This field MUST be set to the number of Command Class entries present in this command (represented by N in the command structure).

Command Class (8 or 16 bits)

CC:0053.04.0B.13.001 This field is used to indicate the command class to which the advertised commands belong. This field MAY carry extended Command Classes

Supported Command List Length (8 bits)

This field is used to advertise how many commands are advertised in the current Command Class' list.

CC:0053.04.0B.11.002 This field MUST be set to the number of Supported Commands present in the current Command Class entry (represented by K in the command structure).

Supported Command (K bytes)

This field is used to advertise the list of commands for the given Command Class entry that can be scheduled for the actual Schedule ID Block.

CC:0053.04.0B.11.003 The length of this field in bytes MUST be according to the corresponding Supported Command List Length field value.

4.89.5 Schedule Set Command

This command is used to create a new schedule or modify an existing schedule.

CC:0053.04.03.11.001 This command MUST enable the schedule if a new schedule is created.

CC:0053.04.03.11.002 This command MUST NOT change the enabled/disabled state of existing schedules

CC:0053.04.03.11.003 If the receiving node supports S0 or S2 Command Class, it MUST NOT process the schedule creation if any of the scheduled command classes is not supported at the security level of the received Schedule Set Command.

CC:0053.04.03.11.004 A controlling node using previous versions may try to schedule unsupported commands. In this case, a receiving node MUST ignore the Schedule Set Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCHEDULE							
Command = SCHEDULE_SET (0x03)							
Schedule ID							
Schedule ID Block							
Start Year							
Recurrence Offset				Start Month			
Res.	Recurrence Mode		Start Day of Month				
Res.	Start Weekday						
Duration Type		Start Hour					
Res.	Relative	Start Minute					
Duration Byte 1 MSB							
Duration Byte 2 LSB							
Reports to Follow [OBSOLETED]							
Number of Cmd to Follow							
Cmd Length 1							
Cmd Byte 1 - 1							
…							
Cmd Byte 1 - N							
…							
Cmd Length P							
Cmd Byte P - 1							
…							
Cmd Byte P - N							

Fields not described below remain unchanged from version 3.

Reports to Follow (8 bits) [OBSOLETED]

CC:0053.04.03.11.005 This field has been OBSOLETED: a sending node MUST use the Transport Service Command Class in order to set a schedule containing commands payload too large to fit in the Z-Wave MAC frame size.

CC:0053.04.03.11.006 This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

4.89.6 Schedule Report Command

The Schedule Report Command is used to advertise the configuration for a specific schedule.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCHEDULE							
Command = SCHEDULE_REPORT (0x05)							
Schedule ID							
Schedule ID Block							
Start Year							
Active_ID/Recurrence Offset (AID_RO)				Start Month			
AID_RO_CTL	Recurrence Mode		Start Day of Month				
Res.	Start Weekday						
Duration Type		Start Hour					
Res.	Relative	Start Minute					
Duration Byte 1							
Duration Byte 2							
Report to Follow [OBSOLETED]							
Number of Cmd to Follow							
Cmd Length 1							
Cmd Byte 1 - 1							
…							
Cmd Byte 1 – N							
…							
Cmd Length P							
Cmd Byte P - 1							
…							
Cmd Byte P - N							

Fields not described below remain unchanged from version 3.

Active_ID/Recurrence Offset (4 bits)

This field is used to advertise the Active_ID or the Recurrence Offset of the advertised schedule.

CC:0053.04.05.11.001 If this field carries the Active_ID value, this field MUST be encoded according to Table 118.

CC:0053.04.05.11.002 If this field carries the Recurrence Offset, this field MUST be encoded according to Table 116 and MUST be interpreted together with the Recurrence Mode field.

Reports to Follow (8 bits) [OBSOLETED]

CC:0053.04.05.11.003 This field has been OBSOLETED: a sending node MUST use the Transport Service Command Class in order to report a schedule containing commands payload too large to fit in the Z-Wave MAC frame size.

CC:0053.04.05.11.004 This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

4.89.7 Schedule State Report Command

This command is used to advertise the status of all schedules supported by the specified schedule block.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCHEDULE							
Command = SCHEDULE_STATE_REPORT (0x09)							
Number of Supported Schedule ID							
Reports to Follow [OBSOLETED]							Override
Active_ID 2				Active_ID 1			
…							
Active_ID N				Active_ID N-1			
Schedule ID Block							

Fields not described below remain unchanged from version 3.

Reports to Follow (7 bits) [OBSOLETED]

CC:0053.04.09.11.001 This field has been OBSOLETED: a sending node MUST use the Transport Service Command Class in order to advertise an Active_ID field too large to fit in the Z-Wave MAC frame size.

CC:0053.04.09.11.002 This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Active_ID (N * 4 bits)

CC:0053.04.09.11.003 The *Active_ID* field MUST be used to advertise the status for each supported schedule ID. Four bits are used for each Schedule ID to indicate the status. This means that :

- Bits 0 to 3 in Active_ID byte 1 represent Schedule ID = 1,.
- Bits 4 to 7 in Active_ID byte 1 represent Schedule ID = 2
- Bits 0 to 3 in Active_ID byte 2 represent Schedule ID = 3
- …

CC:0053.04.09.11.004 The size of this field MUST be the smallest number of bytes needed to advertise the number of supported Schedule IDs advertised in this command.

CC:0053.04.09.11.005 Each 4 bits unit MUST be encoded according to Table 118.

Table 118. Active_ID encoding, version 4

Hex	Description	Description	Version
0x00	Not used	The Schedule ID is not used (not set/configured) or unsupported.	1
0x01	[OBSOLETED] Override + Not used	[OBSOLETED] A sending node MUST NOT use this status and MUST use 0x00 instead (and the override field set to 1 for the Schedule State Report Command).	4
0x02	Not Active	The Schedule ID is used, enabled and currently not active.	1
0x03	Active	The Schedule ID is used, enabled and currently active.	1
0x04	Disabled	The Schedule ID is used and disabled.	1
0x05	Override + Active	The Schedule ID is used, enabled and should currently be active but it is suspended by the Override Schedule The override field MUST be set to 1 when using this status in the Schedule State Report Command	1
0x06	[OBSOLETED] Override + Not Active	[OBSOLETED] A sending node MUST NOT use this status and MUST use 0x02 instead. (and the override field set to 1 for the Schedule State Report Command).	4
0x07	[OBSOLETED] Override + Disabled	[OBSOLETED] A sending node MUST NOT use this status and MUST use 0x04 instead. (and the override field set to 1 for the Schedule State Report Command).	4

CC:0053.04.09.11.006

CC:0053.04.09.11.007

CC:0053.04.09.11.008

CC:0053.04.09.11.009

CC:0053.04.09.11.00A All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

4.90 Schedule Entry Lock Command Class, version 1 [DEPRECATED]

THIS COMMAND CLASS HAS BEEN DEPRECATED

A device MAY implement this command class, but it is RECOMMENDED that new implementations use the Schedule Command Class.
If implementing this command class, it is RECOMMENDED that the Schedule Command Class is also implemented.

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.

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.

4.90.1 Schedule Entry Lock Enable Set Command

This 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)

Table 119, Schedule Entry Lock Enable Set:: Enabled encoding

Value	Description
0x00	Schedule for the user identified is disabled
0x01	Schedule for the user identified is enabled

All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

4.90.2 Schedule Entry Lock Enable All Set Command

This 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.

4.90.3 Schedule Entry Lock Supported Get Command

This command is used to request the number of schedule slots each type of schedule the device supports for every user.

The Schedule Entry Lock Supported Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCHEDULE_ENTRY_LOCK							
Command = SCHEDULE_ENTRY_TYPE_SUPPORTED_GET							

4.90.4 Schedule Entry Lock Supported Report Command

This 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.

4.90.5 Schedule Entry Lock Week Day Schedule Set Command

This 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)

Table 120, Schedule Entry Lock Week Day Schedule Set::Set Action encoding

Set Action	Description
0	Erase the schedule slot
1	Modify the schedule slot for the identified user

All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

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.

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

4.90.6 Schedule Entry Lock Week Days Schedule Get Command

This command gets a week day schedule slot for a identified user and specified schedule slot ID.

The Schedule Entry Lock Week Days Schedule Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

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*.

4.90.7 Schedule Entry Lock Week Day Schedule Report Command

This 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, the time fields SHOULD be set to 0xFF.

4.90.8 Schedule Entry Lock Year Day Schedule Set Command

This command sets or erases 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)

Table 121, Schedule Entry Lock Year Day Schedule Set:: Set Action encoding

Value	Description
0x00	Erase the schedule slot
0x01	Modify the schedule slot for the identified user

All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

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

4.90.9 Schedule Entry Lock Year Day Schedule Get Command

This command gets a year/day schedule slot for an identified user and specified schedule slot ID.

The Schedule Entry Lock Year Day Schedule Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

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							

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*.

4.90.10 Schedule Entry Lock Year Day Schedule Report Command

This 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 the time fields SHOULD be set to 0xFF.

4.91 Schedule Entry Lock Command Class, version 2 [DEPRECATED]

THIS COMMAND CLASS HAS BEEN DEPRECATED

A device MAY implement this command class, but it is RECOMMENDED that new implementations use the Schedule Command Class.
If implementing this command class, it is RECOMMENDED that the Schedule Command Class is also implemented.

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**.**

4.91.1 Schedule Entry Lock Time Offset Get Command

This command is used to request time zone offset and daylight savings parameters.

The Schedule Entry Lock Time Offset Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCHEDULE_ENTRY_LOCK							
Command = SCHEDULE_ENTRY_LOCK_TIME_OFFSET_GET							

4.91.2 Schedule Entry Lock Time Offset Set Command

This 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.

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)

This field MUST specify the number of minutes the time is to be adjusted when daylight savings mode is enabled.

4.91.3 Schedule Entry Lock Time Offset Report Command

This command is used to advertise the 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_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.

4.92 Schedule Entry Lock Command Class, Version 3 [DEPRECATED]

THIS COMMAND CLASS HAS BEEN DEPRECATED

A device MAY implement this command class, but it is RECOMMENDED that new implementations use the Schedule Command Class.
If implementing this command class, it is RECOMMENDED that the Schedule Command Class is also implemented.

The Schedule Entry Lock Command Class provides a scheduling type alongside 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.

4.92.1 Schedule Entry Type Supported Report Command

This 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_ENTRY_LOCK							
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 to 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 to 255 that represents 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 to 255 that represents how many different schedule slots are supported for every user in the system for type Daily Repeating Day.

4.92.2 Schedule Entry Lock Daily Repeating Set Command

This command is used 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)

Table 122, Schedule Entry Lock Daily Repeating Set:: Set Action encoding

Value	Description
0x00	Erase the schedule slot
0x01	Modify the schedule slot for the identified user

All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

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.

Table 123, Schedule Entry Lock Daily Repeating Set:: Week Day Bitmask encoding

Bit	7	6	5	4	3	2	1	0
Value	Res	Sat	Fri	Thr	Wed	Tue	Mon	Sun

The 'Res' bit is reserved and MUST be set to zero by a sending node. Reserved bits MUST be ignored by a receiving node.

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.

4.92.3 Schedule Entry Lock Daily Repeating Get Command

This command is used to request a daily repeating schedule slot for a identified user and specified schedule slot ID.

The Schedule Entry Lock Daily Repeating Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

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*.

4.92.4 Schedule Entry Lock Daily Repeating Report

This command is used to return 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.

4.93 Screen Attributes Command Class, version 1

The Screen Attribute Command Class is 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 may be located on any device in the network.

4.93.1 Screen Attributes Get Command

This command is used to request the screen attributes.

The Screen Attributes Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCREEN_ATTRIBUTES							
Command = SCREEN_ATTRIBUTES_GET							

4.93.2 Screen Attributes Report Command

This command is used to advertise the screen attributes.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCREEN_ATTRIBUTES							
Command = SCREEN_ATTRIBUTES_REPORT							
Reserved			Number of Lines				
Characters per Line							
Line Buffer Size							
Character Encoding							

Reserved

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Number of Lines (5 bits)

Number of lines the screen supports (1..16).

Characters per Line (8 bits)

Number of characters the screen supports on each line (1..255).

Line Buffer Size (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.

Character Encoding (8 bits)

The screen supports the following numerical representations of a character:

Table 124, Screen Attributes Report::Character Encoding encoding

Bit Map	Description
Bit 0	Supports ASCII codes if the bit is 1 and the opposite if 0. See APPENDIX A (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 A
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 A (undefined values are ignored)

All other bits are reserved and MUST be set to zero by a sending node. Reserved bits MUST be ignored by a receiving node.

4.94 Screen Attributes Command Class, version 2

The Screen Attribute Command Class, version 2 introduces the Screen Timeout of the Screen Attributes Command.

Details not mentioned remain the same as in version 1**.**

4.94.1 Screen Attributes Report Command

This command is used to advertise the screen attributes.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCREEN_ATTRIBUTES							
Command = SCREEN_ATTRIBUTES_REPORT							
Reserved		Escape Sequence	Number of Lines				
Characters per Line							
Line Buffer Size							
Character Encoding							
Screen Timeout							

Reserved

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Escape Sequence(1 bit)

If set to 0, escape sequences are not supported by the device.
If set to 1, escape sequences are supported by the device.

Number of Lines (5 bits)

Number of lines the screen supports (1..16).

Characters per Line (8 bits)

Number of characters the screen supports on each line (1..255).

Line Buffer Size (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.

Character Encoding (8 bits)

This field MUST be encoded according to Table 124

All other bits are reserved and MUST be set to zero by a sending node. Reserved bits MUST be ignored by a receiving node.

Screen Timeout (8 bits)

If Screen Timeout is set to 0, the display is always on. A value larger than 0 MUST specify the display timeout in seconds.

4.95 Screen Meta Data Command Class, version 1

The Screen Meta Data Command Class is 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.

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 command longer than two frames, 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.

- 40 kbit/s: At least 35 ms if sending more than 2 frames back-to-back
- 100 kbit/s: At least 15 ms if sending more than 2 frames back-to-back

4.95.1 Screen Meta Data Get Command

This command is used to request the Screen Meta Data Report Command. The Screen Meta Data Get Command is used as handshake to avoid buffer overflow in the receiving node. The Screen Meta Data Get Command will optionally be able to request multiple Screen Meta Data Report Commands to improve the effective bandwidth.

The Screen Meta Data Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCREEN_MD							
Command = SCREEN_MD_GET							
Number of Reports							
NodeID							

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.

NodeID (8 bits)

The NodeID (1..232) specifies the device to receive the requested reports. In case NodeID is equal to 0x00, the information is requested by the source NodeID of the Screen Meta Data Get Command.

4.95.2 Screen Meta Data Report Command

This command is used to send data to the device hosting the screen.

The size of the payload SHOULD NOT be bigger than 48 bytes. 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	Reser-ved	Screen Settings			Character Encoding		
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.

Reserved

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Screen Settings (3 bits)

This field MUST comply with Table 125:

Table 125, Screen Meta Data Report::Screen Settings encoding

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

All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

Character Encoding (3 bits)

This field MUST comply with Table 126:

Table 126, Screen Meta Data Report::Character Encoding encoding

Character Encoding	Description
0	Using standard ASCII codes, see APPENDIX A (values 128-255 are ignored)
1	Using standard ASCII codes and OEM Extended ASCII codes, see APPENDIX A
2	Unicode UTF-16
3	Using standard ASCII codes and Player codes, see APPENDIX A (undefined values are ignored)

All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

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.

Line Settings (3 bits)

This field MUST comply with Table 127:

Table 127, Screen Meta Data Report::Line Settings encoding

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

All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

Clear (1 bit)

Determine if the characters are written directly or line is cleared first.

Table 128, Screen Meta Data Report::Clear encoding

Clear	Description
0	Characters are written directly
1	Line is cleared before characters are written

All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

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 may 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 (N bytes)

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.

4.96 Screen Meta Data Command Class, version 2

The Screen Meta Data Command Class, version 2 introduces a Screen Timeout bit. The support for the Screen Timeout bit may be advertised by the Screen Attribute Command Class, version 2.

Details not mentioned remain the same as in version 1**.**

4.96.1 Screen Meta Data Report Command

This command is used to transfer data to the device hosting the screen. The size of the payload MUST NOT be bigger than 48 bytes. 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			Character Encoding		
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 MUST be encoded according to Table 125.

All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

Character Encoding (3 bits)

The Character Encoding identifier MUST be encoded according to Table 126.

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.

All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

Line Settings (3 bits)

The line settings identifier MUST be encoded according to Table 129.

Table 129, Screen Meta Data Report version 2::Line Settings encoding

Line Settings	Description	Version
0	Characters are written in selected font	1
1	Characters are written as highlighted	1
2	Characters are written using a larger font compared to line settings equal to 0	1
3	Characters are written using a larger font (font B) & highlighted	2
4	Characters are written in selected font (font A), no scroll	2
5	Characters are written in selected font (font A) & highlighted, no scroll	2
6	Characters are written using a larger font (font B), no scroll	2
7	Characters are written using a larger font (font B) & highlighted, no scroll	2

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. Refer to Table 128

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 MAY 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 (N bytes)

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 preset timeout SHOULD be used.

If set to 1 the device SHOULD keep the display powered. This does not affect the RF.

Reserved

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

4.97 Sensor Configuration Command Class, version 1 [OBSOLETED]

THIS COMMAND CLASS HAS BEEN OBSOLETED

New implementations MUST NOT use the Sensor Configuration Command Class. Refer to the Configuration Command Classes.

The 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 may be configured are the same types as the values specified in the Multilevel Sensor Command Class

Most movement sensors may be configured to "ignore" movement if it is not dark. Typically this is done manually. With the Sensor Configuration Command Class this may be configured remotely in a standardised way.

A device supporting the Sensor Configuration Command Class may 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.

4.97.1 Sensor Trigger Level Set Command

This command is 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 types and values MUST be mapped directly from the Multilevel Sensor Command Class.

It is RECOMMENDED that all combinations of precision, scale and size parameters are supported. The Set command MUST support same format of precision, scale and size parameters as can be returned in the report command.

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 1							
…							
Trigger Value N							

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

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

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. Scales are defined by the Z-Wave Alliance.

Size (3 bits)

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

Sensor Type (8 bits)

The Sensor 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 (N bytes)

Refer to the Multilevel Sensor Report Command for information on what trigger values to set.

4.97.2 Sensor Trigger Level Get Command

This command can request the stored trigger level.

The Sensor Trigger Level Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SENSOR_CONFIGURATION							
Command = SENSOR_TRIGGER_LEVEL_GET							

4.97.3 Sensor Trigger Level Report Command

This 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 1							
…							
Trigger Value N							

For fields’ description, refer to the Sensor Trigger Level Set Command.

4.97.4 Mapping example

The report structure of 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 node 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							

4.98 Simple AV Control Command Class, version 1-4

The Simple AV Control Command Class is 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.

4.98.1 Simple AV Control Set Command

This command is 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		
Reserved [OBSOLETED]							
Reserved [OBSOLETED]							
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 node 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:

Table 130, Simple AV Control Set::Key Attributes encoding

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.

All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

Un-supported key attribute values MUST be ignored.

Reserved

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Command MSB, Command LSB (N * 16 bits)

This field is used to carry the AV Control command.

Each 16 bits-unit MUST carry a defined AV Control code. Refer to [19] for the defined AV Control codes. Values not defined in [19] are reserved and MUST NOT be used.

It is possible to specify a sequence of AV commands in one frame. If an AV control command is not supported, it MUST be ignored by a receiving node.

Command numbers 1 through 40 are the most popular commands used in AV remotes.

Command numbers 41 through 363 are less popular and is sorted in alphanumerical order.

Support for Windows Vista Media Center and Media Center 2005 remote controls is added with command numbers from 364 to 377 including 16, 200 and 231.

4.98.2 Simple AV Control Get Command

This command is used to request the number of reports necessary to report the supported AC Commands from the device.

The Simple AV Control Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SIMPLE_AV_CONTROL							
Command = SIMPLE_AV_CONTROL_GET							

4.98.3 Simple AV Control Report Command

This command is used to report the necessary number of reports 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_REPORT							
Number of reports							

Number of reports (8 bits)

The number of reports necessary to report the entire list of supported AC Commands.

4.98.4 Simple AV Control Supported Get Command

This command is used to request the AV Commands supported by the AV device.

The Simple AV Control Supported Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

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.

4.98.5 Simple AV Control Supported Report Command

This command is used 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_SUPPORTED_REPORT							
Report No							
Bit Mask 1							
…							
Bit Mask N							

Report No (8 bits)

Report no. field specify the request report number.

Bit Mask (N bytes)

The Bit Mask fields describe the supported AV Control Commands by the device.

- Bit 0 in Bit Mask 1 indicates if Command #1 is supported.
- Bit 1 in Bit Mask 1 indicates if Command #2 is supported.
- …

If a Command is supported, the bit MUST be set to 1. If a Command is not supported, the bit MUST be set to 0. 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 MUST be determined from the length field in the frame.

4.99 Sound Switch Command Class, version 1

The Sound Switch Command Class is used to manage nodes with a speaker or sound notification capability. It can be used for a doorbell, alarm clock, siren or any device issuing sound notifications.

4.99.1 Terminology

A Sound Switch is a sound notification device with pre-recorded tones. A tone can be a short sound effect as well as an entire song. The volume setting can be configured and the node will subsequently play tones using the configured volume for any tone. A default tone can also be configured, allowing the Sound Switch to play same sound when controlled by a node without the ability to select between tones.

A Sound Switch node advertises a name and duration for each tone. This information is used to guide an end-user in its tone selection and helps controlling nodes to know when the tone is finished playing.

4.99.2 Compatibility considerations

The Sound Switch Command Class is an actuator control Command Class. Refer to section 3.6.

CC:0079.01.00.21.001 A node supporting this Command Class MUST support at least 1 tone.

CC:0079.01.00.21.002 The implemented tone identifier values MUST be in a sequence starting from 1, i.e. a node supporting 10 tones MUST accept Tone Identifiers in the range 1..10.

4.99.3 Sound Switch Tones Number Get Command

This command is used to request the number of tones supported by the receiving node.

CC:0079.01.01.11.001 The Sound Switch Tones Number Report Command MUST be returned in response to this command.

CC:0079.01.01.11.002 This command MUST NOT be issued via multicast addressing.

CC:0079.01.01.11.003 A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SOUND_SWITCH							
Command = SOUND_SWITCH_TONES_NUMBER_GET (0x01)							

4.99.4 Sound Switch Tones Number Report Command

This command is used to advertise the number of tones supported by the sending node.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SOUND_SWITCH							
Command = SOUND_SWITCH_TONES_NUMBER_REPORT (0x02)							
Supported Tones							

Supported Tones (8 bits)

CC:0079.01.02.11.001 This field MUST be set to the total amount of supported tones.

CC:0079.01.02.11.003 This field MUST be in the range 1..254.

4.99.5 Sound Switch Tone Info Get Command

This command is used to query the information associated to a tone at a supporting node.

CC:0079.01.03.11.001 The Sound Switch Tone Info Report Command MUST be returned in response to this command.

CC:0079.01.03.11.002 This command MUST NOT be issued via multicast addressing.

CC:0079.01.03.11.003 A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SOUND_SWITCH							
Command = SOUND_SWITCH_TONE_INFO_GET (0x03)							
Tone Identifier							

Tone Identifier (8 bits)

This field is used to specify the requested Tone Identifier.

CC:0079.01.03.11.004 The implemented tone identifier values MUST be in a sequence starting from 1, i.e. a node supporting 10 tones MUST accept Tone Identifiers in the range 1..10.

CC:0079.01.03.11.005 A sending node MUST specify a Tone Identifier that is supported by a receiving node, i.e. in the range 1..{Total number of supported tones}.

CC:0079.01.03.11.006 A node receiving this command for an unsupported Tone Identifier or for Tone Identifier 0x00 MUST return a zero duration and a zero length name.

4.99.6 Sound Switch Tone Info Report Command

This command is used to advertise the information associated to a tone at a supporting node.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SOUND_SWITCH							
Command = SOUND_SWITCH_TONE_INFO_REPORT (0x04)							
Tone Identifier							
Tone Duration 1							
Tone Duration 2							
Name Length							
Name 1							
...							
Name N							

Tone Identifier (8 bits)

This field is used to advertise the Tone Identifier for which the associated information is being advertised.

Tone Duration (16 bits)

This field is used to advertise duration of the Tone Identifier.

CC:0079.01.04.11.001 This field MUST indicate the time in seconds it takes to play the actual Tone Identifier.

CC:0079.01.04.11.002 This field MUST be set to 0 if the Tone Identifier is set to 0x00 or a value higher than the total number of supported tones advertised in the Sound Switch Tones Number Report Command.

Name Length (8 bits)

This field indicates the length in bytes of the Name field.

CC:0079.01.04.11.003 This field MUST be in the range 1..255 if the Tone Identifier is supported.

CC:0079.01.04.11.004 This field MUST be set to 0 if the Tone Identifier is set to 0x00 or a value higher than the total number of supported tones advertised in the Sound Switch Tones Number Report Command.

Name (N bytes)

This field is used to indicate the assigned name or label for the actual Tone Identifier.

CC:0079.01.04.11.005 The length of this field in bytes MUST comply with the advertised value in the Name Length field. This field MUST be omitted if the Name Length field is set to 0.

CC:0079.01.04.11.006 The field MUST be formatted as a byte array with no zero termination.

CC:0079.01.04.11.007 The characters MUST be encoded in UTF-8 format.

4.99.7 Sound Switch Configuration Set Command

This command is used to set the configuration for playing tones at the supporting node.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SOUND_SWITCH							
Command = SOUND_SWITCH_CONFIGURATION_SET (0x05)							
Volume							
Default Tone Identifier							

Volume (8 bits)

This field is used to specify the volume at which the node will play tones.

CC:0079.01.05.11.001 The encoding of the Volume field MUST be according toTable 131.

Table 131, Sound Switch Configuration Set::Volume encoding

Value	Description
CC:0079.01.05.11.002 0 (0x00)	This value MUST indicate an Off/Mute volume setting (0%)
CC:0079.01.05.11.003 1..100 (0x01..0x64)	These values MUST indicate the actual volume setting from respectively 1% to 100%. A supporting node MAY implement fewer than 100 hardware level. In this case, mapping of hardware levels MUST be monotonous. , i.e. a higher value MUST be mapped to either the same or a higher hardware level.
CC:0079.01.05.11.004 255 (0xFF)	This value MUST indicate to restore most recent non-zero volume setting. This value MUST be ignored if the current volume is not zero (0x00). This value MAY be used to set the Default Tone Identifier and do not modify the volume setting

CC:0079.01.05.11.005 All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

Default Tone Identifier (8 bits)

This field is used to specify the Default Tone. This Tone will be played if receiving the Sound Switch Tone Play Set Command for an unsupported value or for the 0xFF value.

CC:0079.01.05.11.006 The value 0x00 MUST indicate that the receiving node MUST NOT update its current default tone and the command is sent to configure the volume only.

CC:0079.01.05.11.007 Values in the range 1..{Total number of supported tones} MUST indicate that the receiving node MUST use the specified Identifier as Default Tone.

CC:0079.01.05.11.008 Values higher than the Total number of supported tones advertised in the Sound Switch Tones Number Report Command MUST be ignored by a receiving node.

4.99.8 Sound Switch Configuration Get Command

This command is used to request the current configuration for playing tones at the supporting node.

CC:0079.01.06.11.001 The Sound Switch Configuration Report Command MUST be returned in response to this command.

CC:0079.01.06.11.002 This command MUST NOT be issued via multicast addressing.

CC:0079.01.06.11.003 A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SOUND_SWITCH							
Command = SOUND_SWITCH_CONFIGURATION_GET (0x06)							

4.99.9 Sound Switch Configuration Report Command

This command is used to advertise the current configuration for playing tones at the sending node.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SOUND_SWITCH							
Command = SOUND_SWITCH_CONFIGURATION_REPORT (0x07)							
Volume							
Default Tone Identifer							

Volume (8 bits)

This field is used to advertise the current volume setting at the sending node.

CC:0079.01.07.11.001 This field MUST be in the range 0..100. Values in the range 0..100 MUST indicate the current volume percentage.

Default Tone Identifier (8 bits)

This field is used to advertise the current configured Default Tone. This Tone Identifier will be played if receiving a Sound Switch Tone Play Set Command for an unsupported value or for the 0xFF value.

CC:0079.01.07.11.002 This field MUST be in the range 1..{Total number of supported Tones}.

4.99.10 Sound Switch Tone Play Set Command

This command is used to instruct a supporting node to play (or stop playing) a tone.

CC:0079.01.08.11.001 The supporting node MUST play the specified tone immediately when receiving this command.

CC:0079.01.08.11.002 A node already playing a tone MUST interrupt the current tone and start playing the tone specified in this command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SOUND_SWITCH							
Command = SOUND_SWITCH_TONE_PLAY_SET (0x08)							
Tone identifier							

Tone Identifier (8 bits)

CC:0079.01.08.11.003 This field is used to specify the tone that the receiving node MUST play.

CC:0079.01.08.11.004 This field MUST be encoded according to Table 132.

Table 132, Sound Switch Tone Play Set::Tone Identifier encoding

	Value	Description
CC:0079.01.08.11.005	0x00	No Tone MUST be played. A supporting node MUST stop playing any tone when receiving this value.
CC:0079.01.08.11.006	0x01..0xFE	A supporting node MUST play the specified Tone identifier using the configured volume setting. A node receiving a non-supported Tone Identifier (higher than the Total number of supported tones) MUST play the default tone using the configured volume setting.
CC:0079.01.08.11.007	0xFF	The supporting node MUST play the default tone using the configured volume setting.

4.99.11 Sound Switch Tone Play Get Command

This command is used to request the current tone being played by the receiving node.

CC:0079.01.09.11.001 The Sound Switch Tone Play Report Command MUST be returned in response to this command.

CC:0079.01.09.11.002 This command MUST NOT be issued via multicast addressing.

CC:0079.01.09.11.003 A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SOUND_SWITCH							
Command = SOUND_SWITCH_TONE_PLAY_GET (0x09)							

4.99.12 Sound Switch Tone Play Report Command

This command is used to advertise the current tone being played by the sending node.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SOUND_SWITCH							
Command = SOUND_SWITCH_TONE_PLAY_REPORT (0x0A)							
Tone Identifier							

Tone Identifier (8 bits)

CC:0079.01.0A.11.001 This field MUST indicate which tone is currently being played by the sending node. This field MUST be in the range 0..{Total number of supported tones}.

CC:0079.01.0A.11.002 The value 0 MUST indicate that no Tone is currently being played.

This field MUST be set to 0x00 by a sending node if the configured volume is 0x00 (muted).

CC:0079.01.0A.11.003 Values in the range 1..{Total number of supported tones} MUST indicate the Tone that is currently being played.

CC:0079.01.0A.11.004 All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

4.100 Sound Switch Command Class, version 2

The Sound Switch Command Class is used to manage nodes with a speaker or sound notification capability. It can be used for a doorbell, alarm clock, siren or any device issuing sound notifications.

4.100.1 Compatibility Considerations

The Sound Switch Command Class is backwards compatible with the Sound Switch Command Class, version 1.

CC:0079.02.00.21.001 All Commands and fields not mentioned in this version MUST remain unchanged from the Sound Switch Command Class, version 1.

This version extends the following commands to allow a controlling node to modify the volume setting only for a single Tone Play Set Command:

- Sound Switch Tone Play Set Command
- Sound Switch Tone Play Report Command

4.100.2 Sound Switch Tone Play Set Command

This command is used to instruct a supporting node to play (or stop playing) a tone.

CC:0079.02.08.11.001 The supporting node MUST play the specified tone immediately when receiving this command.

CC:0079.02.08.11.002 A node already playing a tone MUST interrupt the current tone and start playing the tone specified in this command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SOUND_SWITCH							
Command = SOUND_SWITCH_TONE_PLAY_SET (0x08)							
Tone identifier							
Play Command Tone Volume							

CC:0079.02.08.11.003 All fields not described below MUST remain unchanged from version 1.

Play Command Tone Volume (8 bits)

CC:0079.02.08.11.004 This field is used to configure the volume for the actual Play Command.

CC:0079.02.08.11.005
CC:0079.02.08.11.006 The volume setting configured with the Sound Switch Configuration Set Command MUST remain unchanged by this command and this field MUST be used for the volume of the actual tone that will start playing.

CC:0079.02.08.11.007 This field MUST be set to 0x00 and ignored by a receiving node if the *Tone identifier* field is set to 0x00.

CC:0079.02.08.11.008 The encoding of this field MUST be according to Table 133.

Table 133, Sound Switch Tone Play Set::Play Command Volume encoding

Value	Description
0 (0x00)	This value MUST indicate to use the configured current volume at the node (Sound Switch Configuration Set Command)
1..100 (0x01..0x64)	These values MUST indicate the actual volume setting from respectively 1% to 100% to use for the Play Set Command. This value SHOULD be used only for critical applications (e.g. security alarm) where it is important to have the sound played even when the device is muted.
255 (0xFF)	This value MUST indicate to use most recent non-zero volume setting if the current volume is muted (0x00). If the current configured volume is not muted (>0x00), this value MUST indicate to use the configured current volume at the node (Sound Switch Configuration Set Command) This value SHOULD be used only for critical application (e.g. security alarm) where it is important to have the sound played even when the device is muted.

All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

4.100.3 Sound Switch Tone Play Report Command

This command is used to advertise the current tone being played by the sending node.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SOUND_SWITCH							
Command = SOUND_SWITCH_TONE_PLAY_REPORT (0x0A)							
Tone Identifier							
Play Command Tone Volume							

All fields not described below MUST remain unchanged from version 1.

Play Command Tone Volume (8 bits)

CC:0079.02.0A.11.001 This field MUSTadvertise the actual playing volume represented with the value from 0x00 …0x64 (0 … 100%). The values 0x00 and 0x64 MUST represent mute and maximum volume, respectively.

This field MUST be set to 0x00 if the Tone Identifier field is set to 0x00.

4.101 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.

4.101.1 Tariff Table Supplier Set Command

This 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							

Utility Timestamp Year (16 bits)

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 bits)

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 bits)

Tariff applies from the specified day of the month between 01 and 31.

Utility Timestamp Hour Local Time (8 bits)

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 bits)

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 bits)

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 (3 bytes)

ISO 4217 defines the currency code. In the table below are some examples of the codes listed:

Table 134, Tariff Table Supplier Set::Currency encoding examples

Currency Code	Currency 1	Currency 2	Currency 3
Pound sterling	G	B	P
US Dollar	U	S	D

Standing Charge Period (5 bits)

This field indicates the stated period that standing charge applies e.g. 50p/week.

Table 135, Tariff Table Supplier Set::Standing Charge Period encoding

Period	Value
Weekly	0x01
Monthly	0x02
Quarterly	0x03
Yearly	0x04
Reserved	0x05-0x1F

Standing Charge 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.

Standing Charge Value (32 bits)

The Standing Charge value MUST be encoded as a 32-bit signed integer. The first byte MUST be the most significant byte. Table 10, Signed field encoding (two's complement representation) shows signed decimal values together with their hexadecimal equivalents.

Reserved

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Number of Supplier Characters (5 bits)

Number of characters defining the name of the utility supplier ID (1 … 32).

Supplier Character (N bytes)

The supplier character fields hold the string identifying the utility supplier. The character presentation uses standard ASCII codes (values 128-255 are ignored).

4.101.2 Tariff Table Set Command

This command adds 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 bits)

The Rate Parameter Set ID indicates the requested parameter set.

Reserved

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Tariff 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.

Tariff Value (32 bits)

The Tariff value is a 32 bit signed field. The first byte is the most significant byte. shows signed decimal values together with their hexadecimal equivalents.

4.101.3 Tariff Table Remove Command

This 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

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Rate Parameter Set IDs (6 bits)

The rate parameter set id's indicates the number of rate parameter set id's in the command.

Rate Parameter Set ID (N bytes)

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.

4.102 Tariff Table Monitor Command Class, version 1

The Tariff Table Monitor Command Class defines the cost for a range of rates.

4.102.1 Tariff Table Supplier Get Command

This command is used to request the name of the utility supplier.

The Tariff Table Supplier Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_TARIFF_TBL_MONITOR							
Command = TARIFF_TBL_SUPPLIER_GET							

4.102.2 Tariff Table Supplier Report Command

This command is used to advertise 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_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							

Refer to description of fields under the Tariff Table Supplier Set Command (section 4.101.1)

4.102.3 Tariff Table Get Command

This command is used to request the tariff for the corresponding rate parameter set.

The Tariff Table Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

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 bits)

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.

4.102.4 Tariff Table Report Command

This commandis used to advertise information relating to a given Rate Parameter Set Identifier.

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 4.101.2)

4.102.5 Tariff Table Cost Get Command

This command is used to request the cost according to rate parameter set ID, rate type, dataset mask and time interval.

The Tariff Table Cost Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

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 bits)

The Rate Parameter Set ID indicates the requested parameter set. Rate Parameter Set ID equal to 0xFF returns overall accumulated cost.

Start Year (16 bits)

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

Start Month (8 bits)

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

Start Day (8 bits)

Specify the day of the month between 01 and 31.

Start Hour Local Time (8 bits)

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

Start 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.

Stop Year (16 bits)

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 bits)

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 bits)

Specify the day of the month between 01 and 31. Setting the parameter to 0xFF indicates now.

Stop Hour Local Time (8 bits)

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 bits)

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 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. Setting the parameter to 0xFF indicates now.

4.102.6 Tariff Table Cost Report Command

This command is used to report a number of time stamped values (historical) in physical units in the device.

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 bits)

The Rate Parameter Set ID indicates the requested parameter set. Rate Parameter Set ID equal to 0xFF returns accumulated cost.

Reserved

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Rate Type (2 bits)

Rate Type specifies the type of parameters in the report. This field MUST be encoded according to Table 69.

Start/Stop Year (16 bits)

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

Start/Stop Month (8 bits)

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 bits)

Specify the day of the month between 01 and 31.

Start/Stop Hour Local Time (8 bits)

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

Start/Stop 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.

Currency (3 bytes)

ISO 4217 defines the currency code. Examples are given in Table 134.

Cost 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.

Cost Value (32 bits)

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.

4.103 Thermostat Fan Mode Command Class, version 1

The Thermostat Fan Mode Command Class, version 1 used for the HVAC's systems manual fan.

4.103.1 Thermostat Fan Mode Set Command

This 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							
Reserved				Fan Mode			

Fan Mode (8 bits)

This field MUST comply with the values indicated for version 1 in Table 136.

Reserved

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

4.103.2 Thermostat Fan Mode Get Command

This command is used to request the fan mode in the device.

The Thermostat Fan Mode Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_FAN_MODE							
Command = THERMOSTAT_FAN_MODE_GET							

4.103.3 Thermostat Fan Mode Report Command

This 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 (8 bits)

Refer to description under 4.103.1 Thermostat Fan Mode Set Command.

Reserved

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

4.103.4 Thermostat Fan Mode Supported Get Command

This command is used to request the supported fan modes from the device.

The Thermostat Fan Mode Supported Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_FAN_MODE							
Command = THERMOSTAT_FAN_MODE_SUPPORTED_GET							

4.103.5 Thermostat Fan Mode Supported Report Command

This command is used to report 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_REPORT							
Bit Mask 1							
…							
Bit Mask N							

Bit Mask (N bytes)

The Bit Mask fields describe the supported fan modes by the thermostat.

- Bit 0 in Bit Mask 1 indicates if Fan Mode = 0 (Auto / Auto Low) is supported.
- Bit 1 in Bit Mask 1 indicates if Fan Mode = 1 (On / On Low) is supported.
- …

If a Fan Mode is supported the bit MUST be set to 1. If a Fan Mode is not supported the bit MUST be set to 0. 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 MUST be determined from the length field in the frame.

4.104 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).

4.104.1 Thermostat Fan Mode Set Command

This 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)

This field MUST comply with the values indicated for version 2 and older in Table 136.

Reserved

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

4.104.2 Thermostat Fan Mode Report Command

This 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

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

4.105 Thermostat Fan Mode Command Class, Version 3

The Thermostat Fan Mode Command Class, version 3 is used for the HVAC’s systems manual fan.

4.105.1 Thermostat Fan Mode Set Command

This 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

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

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.

Fan Mode (4 bits)

This field MUST comply with the values indicated for version 3 and older in Table 136.

4.105.2 Thermostat Fan Mode Get Command

This command is used to request the fan mode in the device.

The Thermostat Fan Mode Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_FAN_MODE							
Command = THERMOSTAT_FAN_MODE_GET							

4.105.3 Thermostat Fan Mode Report Command

This 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 bits)

Refer to description under 4.105.1 Thermostat Fan Mode Set Command.

Reserved

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

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”

4.106 Thermostat Fan Mode Command Class, Version 4-5

The Thermostat Fan Mode Command Class is an extension to support control and status monitoring functions of air-conditioning devices in order to achieve a global framework that covers the majority of

generic functions implemented by world-wide air-conditioning manufacturer. The new features comprises of new Thermostat Fan Modes.

4.106.1 Thermostat Fan Mode Set Command

This 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

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Off (1 bit)

“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 may ignore the off bit.

Fan Mode (4 bits)

If the device transmitting the Thermostat Fan Mode Set command attempts to set a non-supported mode, the receiving thermostat device MUST ignore the command.

Table 136, Thermostat Fan Mode Set version 4::Fan Mode encoding

Fan Mode (4 bits)		Description	CC Version
0x00	AUTO LOW	Will turn the manual fan operation off unless turned on by the manufacturer specific “auto low” algorithms	1
0x01	LOW	Will turn the manual fan operation on. Low speed is selected.	1
0x02	AUTO HIGH	Will turn the manual fan operation off unless turned on by the manufacturer specific “auto high” algorithms	1
0x03	HIGH	Will turn the manual fan operation on. High speed is selected.	1
0x04	AUTO MEDIUM	Will turn the manual fan operation off unless turned on by the manufacturer specific “auto medium” algorithms	2
0x05	MEDIUM	Will turn the manual fan operation on. Medium speed is selected.	2
0x06	CIRCULATION	Will turn the manual fan operation off unless turned on by the manufacturer specific circulation algorithms	3
0x07	HUMIDITY CIRCULATION	Will turn the manual fan operation off unless turned on by the manufacturer specific “humidity circulation” algorithms	3
0x08	LEFT & RIGHT	Will turn the manual fan operation off unless turned on by the manufacturer specific “left & right” circulation algorithms	4
0x09	UP & DOWN	Will turn the manual fan operation off unless turned on by the manufacturer specific “up & down” circulation algorithms	4
0x0A	QUIET	Will turn the manual fan operation off unless turned on by the manufacturer specific “quiet” algorithms	4
0x0B	EXTERNAL CIRCULATION	Will turn the manual fan operation off unless turned on by the manufacturer specific circulation algorithms. This mode will circulate fresh air from the outside.	5
0x0C.. 0x0F	Reserved	These values/modes are reserved for future use. The values cannot be supported by any device and will be ignored.	-

4.106.2 Thermostat Fan Mode Get Command

This command is used to request the fan mode in the device.

The Thermostat Fan Mode Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_FAN_MODE							
Command = THERMOSTAT_FAN_MODE_GET							

4.106.3 Thermostat Fan Mode Report Command

This 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			

Refer to Thermostat Fan Mode Set command for parameter/field descriptions.

4.106.4 Thermostat Fan Mode Supported Get Command

This command is used to request the supported modes from the device.

The Thermostat Fan Mode Supported Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_FAN_MODE							
Command = THERMOSTAT_FAN_MODE_SUPPORTED_GET							

4.106.5 Thermostat Fan Mode Supported Report Command

This command is used to report the supported thermostat modes from the device.

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 (N bytes)

The Bit Mask fields describe the supported modes by the device (Refer to 4.106.1 Thermostat Fan Mode Set Command).

- Bit 0 in Bit Mask 1 field indicates support for mode = 0 (AUTO LOW)
- Bit 1 in Bit Mask 1 field indicates support for mode = 1 (LOW)
- Bit 2 in Bit Mask 1 field indicates support for mode = 2 (AUTO HIGH)
- …

The mode is supported if the bit is 1 and the opposite if 0. 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 MUST be determined from the length field in the frame.

Example:

To indicate the thermostat device supports AUTO LOW, AUTO HIGH and AUTO MEDIUM, the Thermostat Fan Mode Supported Report command MUST be structured as illustrated below.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_FAN_MODE							
Command = THERMOSTAT_FAN_MODE_SUPPORTED_REPORT							
Bit Mask 1							
0	0	0	1	0	1	0	1

4.107 Thermostat Fan State Command Class, version 1-2

The Thermostat Fan State Command Class is used to obtain the fan operating state of the thermostat.

4.107.1 Compatibility considerations

A device supporting Thermostat Fan State CC, Version 2 MUST support Thermostat Fan State CC, Version 1.

Version 2 adds Fan Operating State identifiers for use in the Thermostat Fan State Report Command.

Commands not described in Version 2 stays unchanged from version 1.

4.107.2 Thermostat Fan State Get Command

This command is used to request the fan operating state from the device.

The Thermostat Fan State Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_FAN_STATE							
Command = THERMOSTAT_FAN_STATE_GET							

4.107.3 Thermostat Fan State Report Command

This command is used to report the fan operating state of the device.

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

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Fan Operating State (4 bits)

The fan operating state identifier MUST comply with Table 137.

Table 137, Thermostat Fan State Report::Fan Operating State encoding

Fan Operating State	Description	CC Version
0	Idle / Off	1
1	Running / Running Low – If device only supports one fan speed, this state is used to report the fan is running. If the device is a multi-speed device, this state is used to report that the fan is running at the low speed.	1
2	Running High	1
3	Running Medium	2
4	Circulation Mode	2
5	Humidity Circulation Mode	2
6	Right – Left Circulation Mode	2
7	Up – Down Circulation Mode	2
8	Quiet Circulation Mode	2

All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

4.108 Thermostat Mode Command Class, version 1-2

The Thermostat Mode Command Class is used to control which mode a thermostat operates.

4.108.1 Interoperability considerations

Most thermostat modes require a supporting node to have an associated setpoint, which can managed with the Thermostat Setpoint Command Class. A node supporting this Command Class SHOULD support the Thermostat Setpoint Command Class.

4.108.2 Thermostat Mode Set Command

This command is used to set the thermostat mode at the receiving node.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_MODE (0x40)							
Command = THERMOSTAT_MODE_SET (0x01)							
Reserved				Mode			

Reserved

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Mode (5 bits)

This field is used to set the thermostat mode at the receiving node.

This field MUST be encoded according to Table 138.

4.108.3 Thermostat Mode Get Command

This command is used to request the current mode set at the receiving node.

The Thermostat Mode Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.

A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_MODE (0x40)							
Command = THERMOSTAT_MODE_GET (0x02)							

4.108.4 Thermostat Mode Report Command

This command is used to report the mode from the device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_MODE (0x40)							
Command = THERMOSTAT_MODE_REPORT (0x03)							
Reserved				Mode			

Reserved

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Mode (5 bits)

This field is used to advertise the current thermostat mode at the receiving node.

This field MUST be encoded according to Table 138.

This field MUST NOT be set to 0x05 (RESUME(ON)) in this command.

4.108.5 Thermostat Mode Supported Get Command

This command is used to request the supported modes of a node.

The Thermostat Mode Supported Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_MODE (0x40)							
Command = THERMOSTAT_MODE_SUPPORTED_GET (0x04)							

4.108.6 Thermostat Mode Supported Report Command

This command is used to advertise which modes are supported by the sending node.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_MODE (0x40)							
Command = THERMOSTAT_MODE_SUPPORTED_REPORT (0x05)							
Bit Mask 1							
…							
Bit Mask N							

Bit Mask (N bytes)

This field advertises the Modes supported by the supporting node.

A supporting node MUST support at least one mode. Modes and their minimum required version are described in Table 138. A node MUST NOT support a mode associated to a newer version than the version it supports.

- Bit 0 in Bit Mask 1 represents Mode = 0x00 (Off).
- Bit 1 in Bit Mask 1 represents Mode = 0x01 (Heat).
- …

If a Mode is supported, the corresponding bit MUST be set to '1'.

If a Mode is not supported, the corresponding bit MUST be set to '0'.

This length of this field MUST be set to the minimum amount of bytes which allows advertising all supported Modes.

4.109 Thermostat Mode Command Class, version 3

The Thermostat Mode Command Class is used to control which mode a thermostat operates.

4.109.1 Compatibility considerations

The Thermostat Mode Command Class, version 3 is backwards compatible with the Thermostat Mode Command Class, version 1-2.

All commands and fields not mentioned in this version MUST remain unchanged from the Thermostat Mode Command Class, version 1-2.

This version introduces:

- A new FULL POWER thermostat mode
- MANUFACTURER SPECIFIC mode

4.109.2 Interoperability considerations

The MANUFACTURER SPECIFIC (proprietary) mode MUST NOT be supported by a node if the functionality can be provided using a thermostat mode defined in this command class.

A node supporting the MANUFACTURER SPECIFIC mode MUST fulfill the following conditions:

- The node MUST support as a minimum two other thermostat modes (e.g. HEAT and COOL).
- If the MANUFACTURER SPECIFIC mode functionality can in part be supported by one or more defined thermostat mode in this command class, the node MUST also support these thermostat modes.
- The MANUFACTURER SPECIFIC mode and all of its associated Manufacturer Data fields MUST be described in the product manual.

4.109.3 Thermostat Mode Set Command

This command is used to set the thermostat mode at the receiving node.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_MODE (0x40)							
Command = THERMOSTAT_MODE_SET (0x01)							
No of Manufacturer Data fields			Mode				
Manufacturer Data 1							
…							
Manufacturer Data N							

No of Manufacturer Data fields (3 bits)

This field is used to advertise the length in bytes of the *Manufacturer Data* field in the command. This field MUST be in the range 0..7.

This field MUST be set to 0 if the *Mode* field is not set to 0x1F (MANUFACTURER SPECIFIC).

This field MUST indicate the length in bytes of the *Manufacturer Data* field if the *Mode* field is set to 0x1F (MANUFACTURER SPECIFIC).

Mode (5 bits)

This field is used to set the thermostat mode at the receiving node.

This field MUST be encoded according to Table 138.

Manufacturer Data (N bytes)

This field is used to provide a configuration for the MANUFACTURER SPECIFIC mode. Refer to 4.109.2 Interoperability considerations.

Table 138, Thermostat Mode Set version 3::Mode encoding

Mode (5 bits)		Description	Requires own Setpoint	CC Version
0x00	OFF	This mode is used to switch off the thermostat.	No	1
0x01	HEAT	This mode is used to use activate heating when the temperature is below the Heating (0x01) setpoint.	Yes	1
0x02	COOL	This mode is used to use activate cooling when the temperature is above the Cooling (0x02) setpoint.	Yes	1
0x03	AUTO	This mode is used to regulate the temperature using heating and cooling when the temperature is outside the range defined by the Heating (0x01) and Cooling (0x02) setpoints.	No	1
0x04	AUXILIARY	This mode is used to use activate heating when the temperature is below the Heating (0x01) setpoint, but using and auxiliary or emergency heat source. For example , a heat pump is not efficient when the outside temperature is too low. The auxiliary heat mode may be activated to use a more efficient secondary heat source.	No	1
0x05	RESUME (ON)	This mode is used resume to last active mode (different than OFF 0x00).	No	1
0x06	FAN	This mode is used to activate fans only and circulate air.	No	1
0x07	FURNACE	This mode is used to activate fans to circulate air and heating or cooling will be activated to regulate the temperature at the Furnace (0x07) setpoint.	Yes	1
0x08	DRY	This mode is used to dehumidify and remove moisture. Heating or cooling will be activated to regulate the temperature at the Dry Air (0x08) setpoint.	Yes	1

Mode (5 bits)		Description	Requires own Setpoint	CC Version
0x09	MOIST	This mode is used to humidify and add moisture. Heating or cooling will be activated to regulate the temperature at the Moist Air (0x09) setpoint.	Yes	1
0x0A	AUTO CHANGEOVER	This mode is used to regulate the temperature at the Auto Changeover (0x0A) setpoint using heating and cooling.	Yes	1
0x0B	ENERGY HEAT	This mode is used to use activate heating when the temperature is below the Energy Save Heating (0x0B) setpoint. The Energy Save Heating (0x0B) setpoint is usually lower than the Heating (0x01) setpoint in order to save energy	Yes	2
0x0C	ENERGY COOL	This mode is used to use activate cooling when the temperature is below the Energy Save Cooling (0x0C) setpoint. The Energy Save Cooling (0x0C) setpoint is usually higher than the Cooling (0x02) setpoint in order to save energy	Yes	2
0x0D	AWAY	This mode is used to regulate the temperature using heating and cooling when the temperature is outside the range defined by the Away Heating (0x0D) and Away Cooling (0x0E) setpoints.	Yes	2
0x0E	Reserved	Reserved	-	3
0x0F	FULL POWER	This mode is used to regulate the temperature at the Full Power (0x0F) setpoint using heating and cooling. This mode is intended to use more energy and speed up the temperature regulation to the desired setpoint.	Yes	3
0x1F	MANUFACTURER SPECIFIC	Reserved for vendor specific thermostat mode	No	3

All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

4.109.4 Thermostat Mode Report Command

This command is used to report the current mode of the device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_MODE (0x40)							
Command = THERMOSTAT_MODE_REPORT (0x03)							
No of Manufacturer Data fields			Mode				
Manufacturer Data 1							
…							
Manufacturer Data N							

No of Manufacturer Data fields (3 bits)

This field is used to advertise the length in bytes of the *Manufacturer Data* field in the command. This field MUST be in the range 0..7.

This field MUST be set to 0 if the *Mode* field is not set to 0x1F (MANUFACTURER SPECIFIC).

This field MUST indicate the length in bytes of the *Manufacturer Data* field if the *Mode* field is set to 0x1F (MANUFACTURER SPECIFIC).

Mode (5 bits)

This field is used to advertise the current thermostat mode at the receiving node.

This field MUST be encoded according to Table 138.

This field MUST NOT be set to 0x05 (RESUME(ON)) in this command.

Manufacturer Data (N bytes)

This field is used to advertise the current configuration for the MANUFACTURER SPECIFIC mode. Refer to 4.109.2 Interoperability considerations.

4.110 Thermostat Operating State Command Class, version 1

The Thermostat Operating State Command Class is used to obtain the operating state of the thermostat.

4.110.1 Thermostat Operating State Get Command

This command is used to request the operating state.

The Thermostat Operating State Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_OPERATING_STATE							
Command = THERMOSTAT_OPERATING_STATE_GET							

4.110.2 Thermostat Operating State Report Command

The Thermostat Operating State Report Command is used to report the operating state.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_OPERATING_STATE							
Command = THERMOSTAT_OPERATING_STATE_REPORT							
Reserved				Operating State			

Reserved

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Operating State (4 bits)

The thermostat operating state identifier MUST comply with Table 139.

4.111 Thermostat Operating State Command Class, version 2

The Thermostat Operating State Command Class is used to obtain the operating state of the thermostat as well as logged operating runtime times of thermostat.

4.111.1 Thermostat Operating State Get

This command gets the operating state of the thermostat.

The Thermostat Operating State Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The

Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_OPERATING_STATE							
Command = THERMOSTAT_OPERATING_STATE_GET							

4.111.2 Thermostat Operating State Report

This command is used to report the operating state.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_OPERATING_STATE							
Command = THERMOSTAT_OPERATING_STATE_REPORT							
Operating State							

Operating State (8 bits)

The thermostat operating state identifier MUST be set according to Table 139.

Table 139, Thermostat Operating State Report version 2::Operating State encoding

Operating State	Description	Version
0x00	Idle	1
0x01	Heating	1
0x02	Cooling	1
0x03	Fan Only	1
0x04	Pending Heat. Short cycle prevention feature used in heat pump applications to protect the compressor.	1
0x05	Pending Cool. Short cycle prevention feature used in heat pump applications to protect the compressor.	1
0x06	Vent/Economizer.	1
0x07	Aux Heating	2
0x08	2nd Stage Heating	2
0x09	2nd Stage Cooling	2
0x0A	2nd Stage Aux Heat	2
0x0B	3rd Stage Aux Heat	2

All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

4.111.3 Thermostat Operating State Logging Supported Get

This command is used to request the operating state logging supported by the device.

The Thermostat Operating State Logging Supported Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_OPERATING_STATE							
Command = THERMOSTAT_OPERATING_STATE_LOGGING_SUPPORTED_GET							

4.111.4 Thermostat Operating State Logging Supported Report

This command is used to report 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_LOGGING_SUPPORTED_REPORT							
Bit Mask 1							
…							
Bit Mask N							

Bit Mask (N bytes)

The Bit Mask fields describe the operating state logging supported by the device.

- Bit 0 in Bit Mask 1 is not allocated to any Operating State and MUST beset to zero.
- Bit 1 in Bit Mask 1 indicates if Operating State = 1 (Heating) log is supported.
- Bit 2 in Bit Mask 1 indicates if Operating State = 2 (Cooling) is supported.
- …

If the Operating State log is supported the bit MUST be set to 1. If the Operating State log is not supported the bit MUST be set to 0. 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 MUST be determined from the length field in the frame.

4.111.5 Thermostat Operating State Logging Get

This command is used to request the operating state logging supported by the device.

The Thermostat Operating State Logging Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

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 (N bytes)

The Bit Mask fields describe the operating state log types to be requested.

- Bit 0 in Bit Mask 1 is not allocated to any Operating State and MUST be set to zero.
- Bit 1 in Bit Mask 1 indicates if Operating State = 1 (Heating) log is supported.
- Bit 2 in Bit Mask 1 indicates if Operating State = 2 (Cooling) is supported.
- …

If the Operating State log is supported the bit MUST be set to 1. If the Operating State log is not supported the bit MUST be set to 0. 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 MUST be determined from the length field in the frame.

4.111.6 Thermostat Operating State Logging Report

This 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 bits)

This value indicates how many report frames left before transferring all of the requested thermostat operating state logs.

Operating State Log Type (N * 4 bits)

The Operating State Log Type indicates the operating state type to be requested.

Usage Today Hours (8 bits)

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 bits)

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 bits)

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.

Usage Yesterday Hours (8 bits)

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.

4.112 Thermostat Setback Command Class, version 1

The Thermostat Setback Command Class is used to change the current state of a non-schedule setback thermostat.

4.112.1 Thermostat Setback Set Command

This command is 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 (2 bits)

The setback type field MUST comply with Table 140

Table 140, Thermostat Setback Set::Setback Type encoding

Value	Description
0x00	No override
0x01	Temporary override
0x02	Permanent override
0x03	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.

Setback State (8 bits)

The Setback State MUST comply with Table 141

Table 141, Thermostat Setback Set::Setback State encoding

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

Reserved values MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

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 SHOULD assume the closest possible state.

4.112.2 Thermostat Setback Get Command

This command is used to request the current state of the thermostat.

The Thermostat Setback Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_SETBACK							
Command = THERMOSTAT_SETBACK_GET							

4.112.3 Thermostat Setback Report Command

This command is 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							

Setback Type (2 bits)

Refer to description under the Thermostat Setback Set Command

Setback State (8 bits)

Refer to description under the Thermostat Setback Set Command.

4.113 Thermostat Setpoint Command Class, version 1-2

The Thermostat Setpoint Command Class is used to configure setpoints for the modes supported by a thermostat.

4.113.1 Interoperability Considerations

It has been found that early implementations of this Command Class specification apply two non-interoperable interpretations of the bit mask advertising the support for specific Setpoint Types.

As a consequence, one may find thermostat products and controller products in the marketplace which implement either of the two bit mask interpretations found in Table 142. The notation x.y indicates Bit Mask byte x, bit y.

Implementations of Thermostat Setpoint Command Class, version 1-2 MUST comply with Interpretation A.

4.113.2 Thermostat Setpoint Set Command

This command is used to specify the target value for the specified Setpoint Type at a supporting node.

A supporting node MUST support the same format of *Precision*, *Scale* and *Size* fields values as it sends in the Thermostat Setpoint Supported Report Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_SETPOINT (0x43)							
Command = THERMOSTAT_SETPOINT_SET (0x01)							
Reserved				Setpoint Type			
Precision			Scale		Size		
Value 1							
..							
Value N							

Reserved

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Setpoint Type (4 bits)

This field is used to specify the setpoint to be set at the receiving node .
The value MUST comply with Table 143.

If a non-supported setpoint type is specified, a receiving node MUST ignore the command.

Precision (3 bits)

This field is used to indicate how many decimal places are included in the *Value* field.
For example, the *Value* field set to 1025 with the *Precision* field set to 2 MUST be interpreted as 10.25.

Scale (2 bits)

This field is used to specify what temperature scale is used for the setpoint value.
The field MUST be encoded according to Table 144.

Size (3 bits)

This field is used to indicate the length in bytes of the *Value* field.
This field MUST be set to 1, 2 or 4.

Value (N bytes)

This field is used to advertise the actual setpoint value to be set at the receiving node.

The length of this field MUST be according to the *Size* field value.

The first byte MUST be the most significant byte.

This field MUST be encoded using signed representation and comply with Table 10, Signed field encoding (two's complement representation).

4.113.3 Thermostat Setpoint Get Command

This command is used to request the target value for a given setpoint type that is currently configured at a supporting node.

The Thermostat Setpoint Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.

A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_SETPOINT (0x43)							
Command = THERMOSTAT_SETPOINT_GET (0x02)							
Reserved				Setpoint Type			

Reserved

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Setpoint Type (4 bits)

This field is used to specify which setpoint is requested.
The value MUST comply with Table 143.

A supporting node receving a supported Setpoint Type value MUST be return the same value in the Thermostat Setpoint Report Command.

A supporting node receiving a non-supported Setpoint Type value MUST return the value 0x00 (N/A) in the Thermostat Setpoint Report Command.

4.113.4 Thermostat Setpoint Report Command

This command is used to advertise the target value for a given setpoint type that is currently configured at the sending node.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_SETPOINT (0x43)							
Command = THERMOSTAT_SETPOINT_REPORT (0x03)							
Reserved				Setpoint Type			
Precision			Scale		Size		
Value 1							
..							
Value N							

Reserved

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Setpoint Type (4 bits)

This field is used to advertise the setpoint advertised by the sending node.
The value MUST comply with Table 143.

If this field is set to 0x00 (N/A), it is RECOMMENDED to set the *Size* field to 1 and the *Value* field to 0.

Precision (3 bits)

This field is used to indicate how many decimal places are included in the *Value* field.
For example, the *Value* field set to 1025 with the *Precision* field set to 2 MUST be interpreted as 10.25.

Scale (2 bits)

This field is used to specify what temperature scale is used for the setpoint value.
The field MUST be encoded according to Table 144.

Size (3 bits)

This field is used to indicate the length in bytes of the *Value* field.
This field MUST be set to 1, 2 or 4.

Value (N bytes)

This field is used to advertise the actual setpoint value set at the sending node.

The length of this field MUST be according to the *Size* field value.

The first byte MUST be the most significant byte.

This field MUST be encoded using signed representation and comply with Table 10, Signed field encoding (two's complement representation).

4.113.5 Thermostat Setpoint Supported Get Command

This command is used to query the supported setpoint types.

The Thermostat Setpoint Supported Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_SETPOINT (0x43)							
Command = THERMOSTAT_SETPOINT_SUPPORTED_GET (0x04)							

4.113.6 Thermostat Setpoint Supported Report Command

This command is used to advertise the supported setpoint types.
This command is known to cause interoperability issues. Refer to section 4.113.1.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_SETPOINT (0x43)							
Command = THERMOSTAT_SETPOINT_SUPPORTED_REPORT (0x05)							
Bit Mask 1							
…							
Bit Mask N							

Bit Mask (N bytes)

This field advertises the supported Setpoint Types at the sending node.

A supporting node MUST support at least one setpoint type. Setpoint types and their minimum required version are described in Table 143. A node MUST NOT support a setpoint type associated to a newer version than the version it supports.

This field MUST be encoded according to Table 142, interpretation A:

- Bit 0 in Bit Mask 1 represents Mode = 0x00 (Off).
- Bit 1 in Bit Mask 1 represents Mode = 0x01 (Heat).
- Bit 2 in Bit Mask 1 represents Mode = 0x02 (Cooling).
- Bit 3 in Bit Mask 1 represents Mode = 0x07 (Furnace).
- …

If a Setpoint Type is supported, the corresponding bit MUST be set to ‘1’.

If a Setpoint Type is not supported, the corresponding bit MUST be set to ‘0’.

This length of this field MUST be set to the minimum amount of bytes which allows advertising all supported Setpoint Types.

4.114 Thermostat Setpoint Command Class, version 3

The Thermostat Setpoint Command Class is used to configure setpoints for the modes supported by a thermostat.

4.114.1 Compatibility considerations

The Thermostat Setpoint Command Class, version 3 is backwards compatible with the Thermostat Setpoint Command Class, version 1-2.

This version introduces:

- Setpoint capability discovery (allowed value range).
- New setpoint types.

4.114.2 Interoperability Considerations

It has been found that early implementations of this Command Class specification apply two non-interoperable interpretations of the bit mask advertising the support for specific Setpoint Types.

As a consequence, one may find thermostat products and controller products in the marketplace which implement either of the two bit mask interpretations found in Table 142. The notation x.y indicates Bit Mask byte x, bit y.

Implementations of Thermostat Setpoint Command Class, version 3 MUST comply with Interpretation A.

Table 142, Thermostat Setpoint Types Bit Mask encoding

Support Bit Mask, Interpretation A	Support Bit Mask, Interpretation B	Setpoint Type Identifier	Description	Version
1.0	1.0	0x00	N/A	-
1.1	1.1	0x01	Heating	1
1.2	1.2	0x02	Cooling	1
	1.3	0x03	N/A	-
	1.4	0x04	N/A	-
	1.5	0x05	N/A	-
	1.6	0x06	N/A	-
1.3	1.7	0x07	Furnace	1
1.4	2.0	0x08	Dry Air	1
1.5	2.1	0x09	Moist Air	1
1.6	2.2	0x0A	Auto changeover	1
1.7	2.3	0x0B	Energy Save Heating	2
2.0	2.4	0x0C	Energy Save Cooling	2
2.1	2.5	0x0D	Away Heating	2
2.2	2.6	0x0E	Away Cooling	3
2.3	2.7	0x0F	Full Power	3

All other bits are reserved and MUST be set to zero by a sending node. Reserved bits MUST be ignored by a receiving node.

4.114.3 Thermostat Setpoint Set Command

This command is used to specify the target value for the specified Setpoint Type at a supporting node.

A supporting node MUST support the same format of *Precision*, *Scale* and *Size* fields values as it sends in the Thermostat Setpoint Supported Report Command or Thermostat Setpoint Capabilities Report Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_SETPOINT (0x43)							
Command = THERMOSTAT_SETPOINT_SET (0x01)							

Reserved	Setpoint Type	
Precision	Scale	Size
Value 1		
..		
Value N		

Reserved

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Setpoint Type (4 bits)

This field is used to specify the setpoint to be set at the receiving node.
The value MUST comply with Table 143.

Table 143, Thermostat Setpoint Set::Setpoint Types

Value	Description	CC Version
0x00	N/A	1
0x01	Heating	1
0x02	Cooling	1
0x03	N/A - reserved	-
0x04	N/A- reserved	-
0x05	N/A- reserved	-
0x06	N/A- reserved	-
0x07	Furnace	1
0x08	Dry Air	1
0x09	Moist Air	1
0x0A	Auto Changeover	1
0x0B	Energy Save Heating	2
0x0C	Energy Save Cooling	2
0x0D	Away Heating	2
0x0E	Away Cooling	3
0x0F	Full Power	3

Values marked as not applicable (N/A) MUST be ignored by a receiving node. All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

Precision (3 bits)

This field is used to specify how many decimal places are included in the *Value* field. For example, the decimal value 1025 with precision 2 MUST be interpreted as 10.25.

Scale (2 bits)

This field is used to specify what temperature scale is used for the setpoint value.
The field MUST be encoded according to Table 144.

Table 144, Thermostat Setpoint Set::Scale encoding

Value	Scale used in Value field
0	Celcius
1	Fahrenheit

All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

Size (3 bits)

This field is used to indicate the length in bytes of the *Value* field.
This field MUST be set to 1, 2 or 4.

Value (N bytes)

This field is used to advertise the actual setpoint value to be set at the receiving node.

The length of this field MUST be according to the *Size* field value.

The first byte MUST be the most significant byte.

This field MUST be encoded using signed representation and comply with Table 10, Signed field encoding (two's complement representation).

4.114.4 Thermostat Setpoint Get Command

This command is used to request the target value for a given setpoint type that is currently configured at a supporting node.

The Thermostat Setpoint Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.

A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_SETPOINT (0x43)							
Command = THERMOSTAT_SETPOINT_GET (0x02)							
Reserved				Setpoint Type			

Reserved

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Setpoint Type (4 bits)

This field is used to specify which setpoint is requested.
The value MUST comply with Table 143.

A supporting node receving a supported Setpoint Type value MUST be return the same value in the Thermostat Setpoint Report Command.

A supporting node receiving a non-supported Setpoint Type value MUST return the value 0x00 (N/A) in the Thermostat Setpoint Report Command.

4.114.5 Thermostat Setpoint Report Command

This command is used to advertise the value of a setpoint type.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_SETPOINT (0x43)							
Command = THERMOSTAT_SETPOINT_REPORT (0x03)							
Reserved				Setpoint Type			
Precision			Scale		Size		
Value 1							
..							
Value N							

Reserved

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Setpoint Type (4 bits)

This field is used to advertise the setpoint advertised by the sending node.
The value MUST comply with Table 143.

If this field is set to 0x00 (N/A), it is RECOMMENDED to set the *Size* field to 1 and the *Value* field to 0.

Precision (3 bits)

This field is used to indicate how many decimal places are included in the *Value* field.
For example, the *Value* field set to 1025 with the *Precision* field set to 2 MUST be interpreted as 10.25.

Scale (2 bits)

This field is used to specify what temperature scale is used for the setpoint value.
The field MUST be encoded according to Table 144.

Size (3 bits)

This field is used to indicate the length in bytes of the *Value* field.
This field MUST be set to 1, 2 or 4.

Value (N bytes)

This field is used to advertise the actual setpoint value set at the sending node.

The length of this field MUST be according to the *Size* field value.

The first byte MUST be the most significant byte.

This field MUST be encoded using signed representation and comply with Table 10, Signed field encoding (two's complement representation).

A receiving node MUST ignore values outside the range advertised in the Thermostat Setpoint Capabilities Report Command for the actual Setpoint type.

4.114.6 Thermostat Setpoint Supported Get Command

This command is used to query the supported setpoint types.

The Thermostat Setpoint Supported Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.

A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_SETPOINT (0x43)							
Command = THERMOSTAT_SETPOINT_SUPPORTED_GET (0x04)							

4.114.7 Thermostat Setpoint Supported Report Command

This command is used to advertise the supported setpoint types.
This command is known to cause interoperability issues. Refer to section 4.114.2.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_SETPOINT (0x43)							
Command = THERMOSTAT_SETPOINT_SUPPORTED_REPORT (0x05)							
Bit Mask 1							
…							
Bit Mask N							

Bit Mask (N bytes)

This field advertises the supported Setpoint Types at the sending node.

A supporting node MUST support at least one setpoint type. Setpoint types and their minimum required version are described in Table 143. A node MUST NOT support a setpoint type associated to a newer version than the version it supports.

This field MUST be encoded according to Table 142, interpretation A:

- Bit 0 in Bit Mask 1 represents Mode = 0x00 (Off).
- Bit 1 in Bit Mask 1 represents Mode = 0x01 (Heat).
- Bit 2 in Bit Mask 1 represents Mode = 0x02 (Cooling).
- Bit 3 in Bit Mask 1 represents Mode = 0x07 (Furnace).
- …

If a Setpoint Type is supported, the corresponding bit MUST be set to ‘1’.

If a Setpoint Type is not supported, the corresponding bit MUST be set to ‘0’.

This length of this field MUST be set to the minimum amount of bytes which allows advertising all supported Setpoint Types.

4.114.8 Thermostat Setpoint Capabilities Get Command

This command is used request the supported setpoint value range for an actual Setpoint Type.

The Thermostat Setpoint Capabilities Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_SETPOINT (0x43)							
Command = THERMOSTAT_SETPOINT_CAPABILITIES_GET (0x09)							
Reserved				Setpoint Type			

Reserved

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Setpoint Type (4 bits)

This field is used to specify which setpoint is requested.
The value MUST comply with Table 143.

A supporting node receving a supported Setpoint Type value MUST be return the same value in the Thermostat Setpoint Capabilities Report Command.

A supporting node receiving a non-supported Setpoint Type value MUST return the value 0x00 (N/A) in the Thermostat Setpoint Capabilities Report Command.

4.114.9 Thermostat Setpoint Capabilities Report Command

This command is used advertise the supported setpoint value range for an actual Setpoint Type.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_SETPOINT (0x43)							
Command = THERMOSTAT_SETPOINT_CAPABILITIES_REPORT (0x09)							
Reserved				Setpoint Type			
Min Value Precision			Min Value Scale		Min Value Size		
Min Value 1							
..							
Min Value N							
Max Value Precision			Max Value Scale		Max Value Size		
Max Value 1							
..							
Max Value N							

Setpoint Type (4 bits)

This field is used to advertise the setpoint advertised by the sending node.
The value MUST comply with Table 143.

If this field is set to 0x00 (N/A), it is RECOMMENDED to set the *Min Value Size* and *Max Value Size* fields to 1 and the *Min Value* and *Max Value* fields to 0.

Min Value Precision (3 bits)

This field is used to indicate how many decimal places are included in the *Min Value* field.
For example, the *Min Value* field set to 1025 with the *Min Value Precision* field set to 2 MUST be interpreted as 10.25.

Min Value Scale (2 bits)

This field is used to specify what temperature scale is used for the setpoint *Min Value* field.
The field MUST be encoded according to Table 144.

Min Value Size (3 bits)

This field is used to indicate the length in bytes of the *Min Value* field.
This field MUST be set to 1, 2 or 4.

Min Value (N bytes)

This field is used to advertise the minimum value that is supported for the actual setpoint at the sending node.

The length of this field MUST be according to the *Min Value Size* field value.

The first byte MUST be the most significant byte.

This field MUST be encoded using signed representation and comply with Table 10, Signed field encoding (two's complement representation).

Max Value Precision (3 bits)

This field is used to indicate how many decimal places are included in the *Max Value* field.
For example, the *Max Value* field set to 1025 with the *Max Value Precision* field set to 2 MUST be interpreted as 10.25.

Max Value Scale (2 bits)

This field is used to specify what temperature scale is used for the setpoint *Max Value* field.
The field MUST be encoded according to Table 144 and this field MUST be set to the same value as the *Min Value Scale* field

Max Value Size (3 bits)

This field is used to indicate the length in bytes of the *Max Value* field.
This field MUST be set to 1, 2 or 4.

Max Value (N bytes)

This field is used to advertise the maximum value that is supported for the actual setpoint at the sending node.

The length of this field MUST be according to the *Max Value Size* field value.

The first byte MUST be the most significant byte.

This field MUST be encoded using signed representation and comply with Table 10, Signed field encoding (two's complement representation).

This field MUST be set to a value greater than the *Min Value* field.

4.115 User Code Command Class, version 1

The User Code Command Class is used to manage User Codes in access control systems.

4.115.1 Interoperability considerations

CC:0063.01.00.32.001 A node supporting the Door Lock Command Class SHOULD reflect user code inputs in the door lock status when relevant. (e.g. when the door becomes unlocked by a User Code input, the Door Lock Mode is also updated to unlocked)

This Command Class can be used in conjunction with Schedule Entry Lock Command Class in order to schedule access for users.

CC:0063.01.00.31.001 A node receiving a User Code Get Command MUST advertise what User Code is set in the User Code Report Command.

It has been found that some version 1 nodes wrongfully report obfuscated User Codes in the User Code Report (e.g. '******').

CC:0063.01.00.32.002 A controlling node SHOULD understand that a code has been set correctly but cannot be read back with such nodes.

4.115.2 User Code Set Command

This command is used to set a User Code at the receiving node.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_USER_CODE (0x63)							
Command = USER_CODE_SET (0x01)							
User Identifier							
User ID Status							
User Code 1							
...							
User Code N							

User Identifier (8 bits)

This field is used to specify the actual User Identifier.

CC:0063.01.01.11.001 The implemented User Identifier values MUST be in a sequence starting from 1, i.e. a node supporting 10 User Identifiers MUST accept values in the range 1..10.

CC:0063.01.01.13.001 A receiving node MAY ignore this field if it only supports one User Code.

CC:0063.01.01.11.002
CC:0063.01.01.12.001 The value 0 MUST indicate that the receiving node MUST set the User Code and User Status for all supported User Identifiers. The value 0 SHOULD be used only with User ID Status set to 0x00.

CC:0063.01.01.11.003 If a non-existing User Identifier (higher than the advertised supported users number in the Users Number Report Command) is specified in this command, a receiving node MUST ignore the command

User ID Status (8 bits)

CC:0063.01.01.11.004 The User ID Status field indicates the status of the User Identifier. This field MUST comply with Table 145.

CC:0063.01.01.12.002 The User ID Status 0xFE SHOULD NOT be used in this command.

Table 145, User Code Set::User ID Status encoding

Value	Description
0x00	Available (not set)
0x01	Occupied
0x02	Reserved by administrator
0xFE	Status not available

CC:0063.01.01.11.005 All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

User Code (N bytes)

This field is used to advertise the User Code to be set for the User Identifier.

CC:0063.01.01.11.006 The length of this field MUST be between 4 and 10 bytes. The field's length MUST be determined using the length of the frame.

CC:0063.01.01.11.007 Each byte in this field MUST be a digit encoded with ASCII representation (from 0x30 to 0x39).

CC:0063.01.01.11.008 A node receiving an invalid User Code MUST ignore the command.

CC:0063.01.01.11.009 The User Code field MUST be set to 0x00000000 (4 bytes) when User ID Status is equal to 0x00.

4.115.3 User Code Get Command

This command is used to request the User Code of a specific User Identifier.

CC:0063.01.02.11.001 The User Code Report Command MUST be returned in response to this command.

CC:0063.01.02.11.002 This command MUST NOT be issued via multicast addressing.

CC:0063.01.02.11.003 A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_USER_CODE (0x63)							
Command = USER_CODE_GET (0x02)							
User Identifier							

User Identifier (8 bits)

This field is used to specify the requested User Identifier.

CC:0063.01.02.11.004 The value 0 MUST NOT be specified in the User Code Get Command.

CC:0063.01.02.12.001 If the specified User Identifier is not supported, a responding node SHOULD return a User Code Report Command with the User ID Status set to 0xFE "Status not available".

4.115.4 User Code Report Command

This command is used to advertise a User Code and its current status.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_USER_CODE (0x63)							
Command = USER_CODE_REPORT (0x03)							
User Identifier							
User ID Status							
User Code 1							
...							
User Code N							

For fields' description, refer to 4.115.2 User Code Set Command.

4.115.5 Users Number Get Command

This command is used to request the number of user codes supported by the receiving node.

CC:0063.01.04.11.001 The Users Number Report Command MUST be returned in response to this command.

CC:0063.01.04.11.002 This command MUST NOT be issued via multicast addressing.

CC:0063.01.04.11.003 A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_USER_CODE (0x63)							
Command = USERS_NUMBER_GET (0x04)							

4.115.6 Users Number Report Command

This command is used to report the number of User Codes supported by the sending node.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_USER_CODE (0x63)							
Command = USERS_NUMBER_REPORT (0x05)							
Supported Users							

Supported Users (8 bits)

This field is used to advertise the number of supported User Codes.

CC:0063.01.05.11.001 This field MUST be set to the total amount of supported User Codes. The value '0' MUST indicate that no User Code is supported by the node.

4.116 User Code Command Class, version 2

The User Code Command Class is used to manage user codes in access control systems.

User Code Command Class, version 2 extends the number of supported users to 65535, introduces new User ID Statuses, keypad modes and a Master Code functionality.

4.116.1 Terminology

User Codes are configured on a supporting node for different User Identifiers. Each User Code is associated to one User Identifier. An actual end user with several User Codes will also have several User Identifiers (one for each User Code).

The entry of a user code can trigger different outcomes, such as opening a door, triggering a notification or ignoring the code. This can be configured with the User ID status.

A supporting node may support one or several Keypad Modes. They allow configuring the keypad to accept, ignore or treat differently keypad input.

A node may support the Master Code functionality. The Master Code is a separate unique code used to get access to the node's administrator functionalities, such as the network settings and/or User Code management. A node may require the Master Code to be always set and in this case advertise that the Master Code can be modified but cannot be deactivated using Z-Wave.

A node may support the User Code Checksum functionality. A controlling node can request a checksum representing all the user code set at the supporting node to ensure that the user code databases are synchronized.

4.116.2 Compatibility considerations

The User Code Command Class, version 2 is backwards compatible with the User Code Command Class, version 1.

CC:0063.02.00.21.001 All commands and fields not mentioned in this version MUST remain unchanged from the User Code Command Class, version 1.

The Extended User Code Set Command permits to configure several user codes per command. The following command has been extended to match the extended number of supported users:

- Users Number Report Command

The following commands are also introduced to address the extended range of users:

- Extended User Code Set Command
- Extended User Code Get Command
- Extended User Code Report Command

Commands for advertising node capabilities, configuring the keypad mode and the Master Code are introduced:

- User Code Capabilities Get Command
- User Code Capabilities Report Command
- User Code Keypad Mode Get Command
- User Code Keypad Mode Report Command
- Master Code Set Command
- Master Code Get Command
- Master Code Report Command

User Code Command Class, version 2 does not extend the existing User Code Set/Get/Report commands from version 1. The first 255 users that can be addressed by the User Code Set/Get
CC:0063.02.00.21.002 Commands MUST be identical to the first 255 users that can be addressed by the (version 2) Extended User Code Set/Get Commands.

CC:0063.02.00.21.003 Addressing User Codes for extended User Identifiers (in the range 256..65535) MUST be done with the (version 2) Extended User Code Set/Get Commands.

CC:0063.02.00.21.004 A version 1 node MAY advertise support for 0 User Codes in the Users Number Report Command.
A version 2 node MUST advertise support for at least 1 User Code.

CC:0063.02.00.21.005 The implemented User Identifier values MUST be in a sequence starting from 1, i.e. a node supporting 10 Users MUST accept User Identifier values in the range 1..10. However, a controlling node MAY assign user codes non-continuously in the range of supported User Identifier.

4.116.3 Interoperability considerations

This Command Class can be used in conjunction with the Entry Control Command Class to report User Code input to a controlling application.

CC:0063.02.00.31.001 A node supporting the “Messaging” User ID Status MUST support the Entry Control Command Class, version 1 or newer or the Notification Command Class, version 8 or newer.

CC:0063.02.00.31.002 A node supporting this Command Class MUST support the Door Lock Command Class and SHOULD support the Timed Operation Type.

CC:0063.02.00.31.003 A supporting node MUST reflect user code inputs in the door lock status when relevant. (e.g. when the door becomes unsecured by a User Code input, the Door Lock Operation mode is updated to unsecure with a timeout)

CC:0063.02.00.33.001 A node supporting this Command Class MAY have an Association group issuing the corresponding Door Lock Operation Set Commands when valid User Codes are input in order to control several door locks simultaneously.

This Command Class can be used in conjunction with Schedule Entry Lock Command Class to schedule access for users.

CC:0063.02.00.31.004 A node receiving a User Code Get Command MUST advertise what User Code is set in the User Code Report Command.

4.116.4 Security considerations

A node identifies users on the background of User Code entry alone without User Identifier, the following security measures should be taken into consideration:

CC:0063.02.00.42.001
- Users SHOULD NOT be allowed to choose their own user codes. Users tend to choose codes such as birthdates and therefore the risk of having duplicate codes is high. A user may learn about another User Code when trying to set his own User Code and receiving a duplicate/rejection message. In such a case, User Codes SHOULD be generated randomly by the administrating company, if applicable.

CC:0063.02.00.42.002
- If a node is configured with a large number of User Codes: (e.g. 500), it is RECOMMENDED to increase the number of digits in each and every user code, making is less likely to randomly guess an assigned user code.

CC:0063.02.00.42.003
- It is RECOMMENDED to implement mechanisms preventing brute force attacks, such as exponential increasing waiting time between attempts or lock out after a number of failed attempts.

CC:0063.02.00.41.001 If a node allows updating User Codes or User ID Statuses locally via a user interface using a Master Code, the supporting node MUST issue User Code Report Commands via the Lifeline to advertise local updates.

4.116.5 User Code Set Command

This command is used to set a User Code at the receiving node.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_USER_CODE (0x63)							
Command = USER_CODE_SET (0x01)							
User Identifier							
User ID Status							
User Code 1							
...							
User Code N							

CC:0063.02.01.11.001 All fields not described below MUST remain unchanged from version 1.

User ID Status (8 bits)

The User ID Status field indicates the state of the User Identifier.

CC:0063.02.01.11.005 This field MUST comply with Table 147, however the requirements in Table 146 MUST take precedence over the requirements in Table 147.

CC:0063.02.01.11.006 If a non-supported User ID Status is specified, a receiving node MUST ignore the command.

User Identifier (8 bits)

This field is used to specify the actual User Identifier.

CC:0063.02.01.11.002 The implemented User Identifier values MUST be in a sequence starting from 1, i.e. a node supporting 10 User Codes MUST accept values in the range 1..10.

CC:0063.02.01.11.003 The value 0 MUST indicate that the receiving node MUST set the User Code and User Status for all supported user identifiers less than 256. A receiving node supporting the User Code Command Class, version 2 MUST NOT apply the status to users from the extended range (User ID > 255) when this field is set to 0.

CC:0063.02.01.12.001 The value 0 SHOULD be used only with User ID Status set to 0x00.

CC:0063.02.01.11.004 If a non-existing User Identifier (higher than the advertised supported users number in the Users Number Report Command) is specified in this command, a receiving node MUST ignore the command

4.116.6 Users Number Report Command

This command is used to report the number of users that the sending node supports.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_USER_CODE (0x63)							
Command = USERS_NUMBER_REPORT (0x05)							
Supported Users							
Extended Supported Users 1 (MSB)							
Extended Supported Users 2 (LSB)							

Supported Users (8 bits)

This field is used to advertise the number of supported users for version 1 controlling nodes.

CC:0063.02.05.11.001 This field MUST be in the range 1..255.

CC:0063.02.05.11.002 If the node implements more than 255 users, this field MUST be set to 255.
If the node implements 255 users or less, this field MUST be set to the total amount of supported users.

Extended Supported Users (16 bits)

This field is used to advertise the number of supported users for version 2 controlling nodes.

CC:0063.02.05.11.003 If the node supports less than 256 users, this field MUST be identical to the Supported Users field.
If the node implements 256 users or more, this field MUST be set to the total amount of supported users.

4.116.7 User Code Capabilities Get Command

This command is used to request the User Code capabilities of a node.

CC:0063.02.06.11.001 The User Code Capabilities Report Command MUST be returned in response to this command.

CC:0063.02.06.11.002 This command MUST NOT be issued via multicast addressing.

CC:0063.02.06.11.003 A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_USER_CODE (0x63)							
Command = USER_CODE_CAPABILITIES_GET (0x06)							

4.116.8 User Code Capabilities Report Command

This command is used to advertise User Code capabilities.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_USER_CODE (0x63)							
Command = USER_CODE_CAPABILITIES_REPORT (0x07)							
MC Support	MCD support	Res	Supported User ID Status Bit Mask Length				
Supported User ID Status Bit Mask 1							
...							
Supported User ID Status Bit Mask N							
UCC Support	MUCR Support	MUCS Support	Supported Keypad Modes Bit Mask Length				
Supported Keypad Modes Bit Mask 1							
...							
Supported Keypad Modes Bit Mask M							
Reserved			Supported Keys Bit Mask Length				
Supported Keys Bit Mask 1							
…							
Supported Keys Bit Mask L							

Reserved / Res

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

MC (Master Code) Support (1 bit)

This field indicates if the sending node supports the Master Code functionality.

CC:0063.02.07.11.001 The value 1 MUST indicate that the Master Code functionality is supported.
The value 0 MUST indicate that the Master Code functionality is not supported.

MCD (Master Code Deactivation) Support (1 bit)

This field indicates if the sending node supports reporting the Master Code can be deactivated using a Master Code Set Command.

CC:0063.02.07.11.002 The value 1 MUST indicate that the Master Code may be deactivated by issuing a Master Code Set with the length field set to 0.
The value 0 MUST indicate that the Master Code cannot be deactivated and Master Code Set commands with the length field set to 0 will be ignored.

CC:0063.02.07.11.003 This field MUST be set to 0 by a sending node if the MC Support field is set to 0.

UCC (User Code Checksum) Support (1 bit)

This field indicates if the sending node supports the User Code Checksum functionality.

CC:0063.02.07.11.004 The value 1 MUST indicate that the User Code Checksum functionality is supported.
The value 0 MUST indicate that the User Code Checksum functionality is not supported.

MUCR (Multiple User Code Report) Support (1 bit)

This field indicates if the sending node supports reporting the Multiple User Codes at once in a single Extended User Code Report Command. This functionality should be supported by node supporting large amount of User Codes.

CC:0063.02.07.11.005 The value 1 MUST indicate that the Multiple User Code Report functionality is supported.
The value 0 MUST indicate that the Multiple User Code Report functionality is not supported.

MUCS (Multiple User Code Set) Support (1 bit)

This field indicates if the sending node supports being set Multiple User Codes at once in a single Extended User Code Set Command. This functionality should be supported by node supporting large amount of User Codes.

CC:0063.02.07.11.006 The value 1 MUST indicate that the Multiple User Code Set functionality is supported.
The value 0 MUST indicate that the Multiple User Code Set functionality is not supported.

Supported User ID Status Bit Mask Length (5 bits)

CC:0063.02.07.11.007 This field MUST advertise the length in bytes of the *Supported User ID Status Bit Mask* field carried in the command.

CC:0063.02.07.11.008 This field MUST be set to the minimum value which allows advertising all supported User ID Statuses.

Supported User ID Status Bit Mask (N bytes)

CC:0063.02.07.11.009 This field advertises the supported User ID status values. The length of this field in bytes MUST match the value advertised in the *Supported User ID Status Bit Mask Length* field.

A node MUST support the User ID Status values 0x00, 0x01 and 0x02.

- Bit 0 in Bit Mask 1 represents User ID status 0x00.
- Bit 1 in Bit Mask 1 represents User ID status 0x01.
- …

User ID Status values are described in Table 147.

CC:0063.02.07.11.00A If a User ID Status is supported, the corresponding bit MUST be set to ‘1’.
If a User ID Status is not supported, the corresponding bit MUST be set to ‘0’.

CC:0063.02.07.11.00B All other bits are reserved and MUST be set to zero by a sending node. Reserved bits MUST be ignored by a receiving node.

Supported Keypad Modes Bit Mask Length (5 bits)

CC:0063.02.07.11.00C This field MUST advertise the length in bytes of the *Supported Keypad Modes Bit Mask* field carried in the command.

CC:0063.02.07.11.00D This field MUST be set to the minimum value which allows advertising all supported keypad modes.

Supported Keypad Mode Bit Mask (M bytes)

CC:0063.02.07.11.00E This field describes the supported keypad mode values. The length of this field in bytes MUST match the value advertised in the *Supported Keypad Modes Bit Mask Length* field.

CC:0063.02.07.11.00F A node MUST support the keypad mode 0x00 (normal mode).

- Bit 0 in Bit Mask 1 represents keypad mode 0x00 (normal mode).
- Bit 1 in Bit Mask 1 represents keypad mode 0x01 (vacation mode).
- …

Keypad mode values are described in Table 146.

CC:0063.02.07.11.010 If a keypad mode is supported, the corresponding bit MUST be set to '1'.
If a keypad mode is not supported, the corresponding bit MUST be set to '0'.

CC:0063.02.07.11.011 All other bits are reserved and MUST be set to zero by a sending node. Reserved bits MUST be ignored by a receiving node.

Supported Keys Bit Mask Length (5 bits)

CC:0063.02.07.11.012 This field MUST advertise the length in bytes of the *Supported Keys Bit Mask* field carried in the command.

CC:0063.02.07.11.013 This field MUST be set to the minimum value which allows advertising all supported ASCII codes.

Supported Keys Bit Mask (L bytes)

CC:0063.02.07.11.014 This field describes the supported keys that can be input on the keypad by end users. The length of this field in bytes MUST match the value advertised in the *Supported Keys Bit Mask Length* field.

CC:0063.02.07.11.015 This field MUST advertise the ASCII codes that represent the supported keys.

- Bit 0 in Bit Mask 1 represents ASCII code 0-
- Bit 1 in Bit Mask 1 represents ASCII code 1.
- …
- Bit 7 in Bit Mask 16 represents ASCII code 127

CC:0063.02.07.11.016 If an ASCII code is supported, the corresponding bit MUST be set to '1'.
If an ASCII code is not supported, the corresponding bit MUST be set to '0'.

CC:0063.02.07.11.017 All other bits are reserved and MUST be set to zero by a sending node. Reserved bits MUST be ignored by a receiving node.

CC:0063.02.07.11.018 A supporting node MUST NOT advertise an ASCII code representing the "enter" button on the keypad, but only ASCII codes that can be part of the user codes.

CC:0063.02.07.12.001 A supporting node SHOULD support the ASCII codes 0x30..0x39 (decimal digits 0..9).

4.116.9 User Code Keypad Mode Set Command

This command is used to set the keypad mode at the receiving node.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_USER_CODE (0x63)							
Command = USER_CODE_KEYPAD_MODE_SET (0x08)							
Keypad Mode							

Keypad Mode (8 bits)

This field is used to set the keypad mode at the receiving node.

CC:0063.02.08.11.001 This field MUST be encoded according to Table 146.

CC:0063.02.08.11.002 A supporting node being set a keypad mode MUST respect the requirements associated to this mode in Table 146. The requirements in Table 146 (keypad mode) MUST take precedence over the requirements in Table 147 (user code statuses).

CC:0063.02.08.11.003 This command MUST be ignored by a receiving node if the specified keypad mode is not supported.

Table 146, User Code Keypad Mode Set::Keypad mode encoding

Value	Description	Version
0x00	Normal mode. This mode MUST be the default mode. When the Normal mode is active, the supporting node: • MUST work normally with all defined User Codes according to Table 147. • MUST accept the Master Code, if supported.	1
0x01	Vacation mode. This mode is used to disable/switch off the User Code functionalities for normal users. When the Vacation mode is active, the supporting node: • MUST ignore all input of assigned User Code (i.e. with Status ID different than 0x00 Available) • MUST react according to Table 147 when being input unknown codes (i.e. with Status ID 0x00 Available). • MUST accept the Master Code, if supported	2
0x02	Privacy mode. This mode is used to completely disable/switch off the User Code functionalities. When the Privacy mode is active, the supporting node: • MUST ignore any keypad input • MUST ignore the Master Code, if supported. This mode can be used e.g. to disallow User Code keypad use completely or to rely only on Entry Control Notifications, if supported.	2
0x03	Locked Out mode. This mode is used as a security measure to prevent brute force attacks. This mode is normally activated by the supporting node itself. If the node activates the Locked Out mode, it MUST issue a User Code Keypad Mode Report Command to the Lifeline destination. When the Locked Out mode is active, the supporting node: • MUST ignore any keypad input • MUST ignore the Master Code, if supported. A supporting node MUST return to the last active mode automatically after a defined timeout if it activated this mode. A controlling node may unlock the keypad again by issuing this command with a different mode.	2

CC:0063.02.08.11.004 All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

4.116.10 User Code Keypad Mode Get Command

This command is used to request a node about its current keypad mode.

CC:0063.02.09.11.001 The User Code Keypad Mode Report Command MUST be returned in response to this command.

CC:0063.02.09.11.002 This command MUST NOT be issued via multicast addressing.

CC:0063.02.09.11.003 A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_USER_CODE (0x63)							
Command = USER_CODE_KEYPAD_MODE_GET (0x09)							

4.116.11 User Code Keypad Mode Report Command

This command is used by a node to advertise its current keypad mode.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_USER_CODE (0x63)							
Command = USER_CODE_KEYPAD_MODE_REPORT (0x0A)							
Keypad Mode							

Keypad Mode (8 bits)

This field is used to advertise the keypad mode currently active at the sending node.

CC:0063.02.0A.11.001 This field MUST be encoded according to Table 146.

4.116.12 Extended User Code Set Command

This command is used to set one or more User Codes in the receiving node.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_USER_CODE (0x63)							
Command = EXTENDED_USER_CODE_SET (0x0B)							
Number of User Codes							
User Identifier 1 (MSB) 1							
User Identifier 2 (LSB) 1							
User ID Status 1							
Reserved				User Code Length 1			
User Code 1, 1							
…							
User Code N, 1							
…							
User Identifier 1 (MSB) M							
User Identifier 2 (LSB) M							
User ID Status M							
Reserved				User Code Length M			
User Code 1, M							
…							
User Code N, M							

Number of User Codes (8 bits)

This field is used to specify how many user codes blocks are contained in the actual command.

CC:0063.02.0B.11.001 This field MUST be in the range 1..255. A node MUST respect the Z-Wave MAC frame size or Transport service limits when sending this command, which means in most cases that this field SHOULD NOT be set to a value higher than 10.

CC:0063.02.0B.11.002 This field MUST be ignored and considered as set to 1 by a receiving node if it advertises no support for Multiple User Code Set (MUCS) in the User Code Capabilities Report Command

CC:0063.02.0B.11.003 The number of User Code blocks contained in the command MUST be according to this field. A User Code block MUST comprise the following fields:

- User Identifier
- User ID Status
- User Code Length
- User Code

User Identifier (16 bits)

This field is used to specify the User Identifier for the actual User Code block.

CC:0063.02.0B.11.004 The first byte MUST carry the most significant byte of the 16-bit value.

CC:0063.02.0B.11.005 The value 0 MUST indicate that the receiving node MUST set the User Code and User Status for all supported User Identifiers. The value 0 SHOULD be used only with User ID Status set to 0x00 (erasing all User Codes).

CC:0063.02.0B.11.006 If a non-existing User Identifier (higher than the advertised supported users number in the Users Number Report Command) is specified in this command, a receiving node MUST ignore the actual User Code block

User ID Status (8 bits)

CC:0063.02.0B.11.007 The User ID Status field indicates the state of the User Identifier for the actual User Code block. This field MUST comply with Table 147, however the requirements in Table 146 MUST take precedence over the requirements in Table 147.

CC:0063.02.0B.11.008 If a non-supported User ID Status is specified, a receiving node MUST ignore the actual User Code block.

Table 147, Extended User Code Set::User ID Status encoding

Value	Description	Version
0x00	Available: The database entry represented by this User Identifier is not in use. This status MUST be used to delete codes or indicate that an identifier is not assigned. When a user enters an unassigned/unknown code, the node: • MUST indicate that the code is not accepted • MUST NOT grant access to this user • SHOULD take preventive actions for further attempts to enter User ID and/or User Code (refer to 4.116.4).	1

Value	Description	Version
0x01	Enabled / Grant Access The database entry represented by this User Identifier is in use and enabled. When a user enters a code with this status, the node: • MUST indicate that the code is accepted • MUST grant access to the user. • MUST secure the Door Lock again after a timeout (either configured if set in Timed Operation or pre-defined if the Door Lock is set in Constant Operation)	1

Value	Description	Version
0x02	Disabled The database entry represented by this User Identifier is in use but disabled. E.g. It allows a controller to push User Codes to a supporting node and let these codes be subsequently activated by a local interface. When a user enters a code with this status, the node: • MUST indicate that the code is not accepted • MUST NOT grant access to the user.	1
0x03	Messaging: The database entry represented by this User Identifier is in use. The value is used to relay notifications to a controlling application. E.g. A user may enter a messaging code for notifying that maintenance started or an application should activate the alarm system. When a user enters a code with this status, the node: • SHOULD indicate that the code is accepted • MUST NOT grant access to this user • MUST NOT take preventive actions for further attempts to enter User ID and/or User Code (refer to 4.116.4). • MUST send an Entry Control Notification or a Notification Report via the Lifeline Group to identify the messaging user code.	2
0x04	Passage Mode: The database entry represented by this User Identifier is in use. The value is used to let the Door Lock permanently grant access. Passage Mode can be activated and deactivated using the Door Lock Operation Set (Unsecured without timeout / Secured) or via a Passage Mode User Code input. E.g. A receptionist comes in the morning and enters a Passage Mode User Code to allow anybody to subsequently enter the premises without entering a code until Passage Mode is deactivated again. When a user enters a code with this status, the node: • MUST indicate that the code is accepted • MUST toggle the Door Lock operation state between secured and unsecured without timeout (and also disable any auto-relock functionality). • MUST send a Door Lock Operation Report and Door Lock Configuration Report via the Lifeline Association Group to advertise the new Door Lock state.	2

All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

Reserved

CC:0063.02.0B.11.009 This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

User Code Length (4 bits)

This field is used to advertise the length in bytes of the corresponding User Code field in the actual User Code block.

CC:0063.02.0B.11.00A The User Code Length MUST be in the range 4..10 if the User ID Status is different than 0x00.

The User Code Length MUST be set to 0 if the User ID Status is set to 0x00.

User Code (N bytes)

This field is used to advertise the User Code to be set for the User ID for the actual User Code block.

CC:0063.02.0B.11.00B The length of this field in bytes MUST be according to the corresponding User Code Length field value. If the User Code Length is set to 0, this field MUST be omitted.

CC:0063.02.0B.11.00C Each byte in this field MUST be encoded with ASCII representation.

CC:0063.02.0B.11.00D A supporting node receiving a non-supported ASCII character MUST ignore the actual User Code block.

A supporting node receiving a User Code identical to one already set to another User ID or identical to the Master Code MUST ignore the actual User Code block.

4.116.13 Extended User Code Get Command

This command is used to request the User Code of a specific User Identifier.

CC:0063.02.0C.11.001 The Extended User Code Report Command MUST be returned in response to this command

CC:0063.02.0C.11.002 This command MUST NOT be issued via multicast addressing.

CC:0063.02.0C.11.003 A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_USER_CODE (0x63)							
Command = EXTENDED_USER_CODE_GET (0x0C)							
User Identifier 1 (MSB)							
User Identifier 2 (LSB)							
Reserved							Report more

User Identifier (16 bits)

This field is used to specify the requested User Identifier.

CC:0063.02.0C.11.004 The first byte MUST carry the most significant byte of the 16 bit value.

This field MUST NOT be set to 0.

CC:0063.02.0C.11.005 A supporting node MUST return this value in the first User Code block of the returned Extended User Code Report Command.

CC:0063.02.0C.11.006 If a non-existing User Identifier (higher than the advertised supported users number in the Users Number Report Command) is specified in this command, a responding node MUST return an Extended User Code Report Command with the User ID Status set to 0xFE: “Status not available”.

Report more (1 bit)

This field is used to instruct the receiving node to report as many User Codes as possible within a single Z-Wave command (that fits in a single Z-Wave frame) because the sending node intends to read the whole (or a large part of the) User Code database.

CC:0063.02.0C.11.007 This field MUST be ignored by a receiving node if it advertises no support for Multiple User Code Report (MUCR) in the User Code Capabilities Report Command.

CC:0063.02.0C.11.008 The value 0 MUST indicate to return a report for the requested User Identifier only.

The value 1 MUST indicate to return a report for the requested User Identifier and additionally report as many User Identifiers as possible in the response.

When this field is set to 1, a node advertising support for Multiple User Code Report (MUCR) in the User Code Capabilities Report Command:

CC:0063.02.0C.11.009
- MUST return at least 2 User Code blocks in the Extended User Code Report Command unless the last used User Identifier or a non-supported User Identifier is requested.

CC:0063.02.0C.12.001
- SHOULD return as many User Code blocks as possible.

CC:0063.02.0C.11.00A
- Subsequent User Code blocks MUST contain consecutive User Identifier having a User ID Status different than 0. For example, a node supporting 4 User IDs that are set as follow MUST report User Identifier 1, 3, and 4 when requested about User Identifier 1:
  - User ID 1, status 0
  - User ID 2, status 0
  - User ID 3, status 1, code 38473
  - User ID 4, status 4, code 9277

4.116.14 Extended User Code Report Command

This command is used to advertise the User Code of a specific User Identifier.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_USER_CODE (0x63)							
Command = EXTENDED_USER_CODE_REPORT (0x0D)							
Number of User Codes							
User Identifier 1 (MSB) 1							
User Identifier 2 (LSB) 1							
User ID Status 1							
Reserved				User Code Length 1			
User Code 1, 1							
…							
User Code N, 1							
…							
User Identifier 1 (MSB) M							
User Identifier 2 (LSB) M							
User ID Status M							
Reserved				User Code Length M			
User Code 1, M							
…							
User Code N, M							
Next User Identifier 1 (MSB)							
Next User Identifier 2 (LSB)							

Number of User Codes (8 bits)

This field is used to specify how many user codes blocks are contained in the actual command.

CC:0063.02.0D.11.001 This field MUST be in the range 1..255. A node MUST respect the Z-Wave MAC frame size or Transport service limits when sending this command, which means in most cases that this field SHOULD NOT be set to a value higher than 8.

CC:0063.02.0D.11.002 The number of User Code blocks contained in the command MUST be according to this field. A User Code block MUST comprise the following fields:

- User Identifier
- User ID Status
- User Code Length
- User Code

CC:0063.02.0D.11.003 This field MUST be set to 1 by a node advertising no support for Multiple User Code Report (MUCR) in the User Code Capabilities Report Command.

User Identifier (16 bits)

This field is used to specify the actual User Identifier for the actual User Code block.
CC:0063.02.0D.11.004 The first byte MUST carry the most significant byte of the 16-bit value.

User ID Status (8 bits)

CC:0063.02.0D.11.005 This field indicates the state of the User Identifier for the actual User Code block. This field MUST comply with Table 148

Table 148, Extended User Code Report::User ID Status encoding

Value	Description	Version
0x00	Available. Refer to Table 147	1
0x01	Enabled / Grant Access. Refer to Table 147	1
0x02	Disabled. Refer to Table 147	1
0x03	Messaging. Refer to Table 147	2
0x04	Passage Mode. Refer to Table 147	2
0xFE	Status not available. The requested User Identifier is not valid (either 0 or higher than the supported users number in the Users Number Report Command)	1

CC:0063.02.0D.11.006 All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

User Code Length (4 bits)

This field is used to advertise the length in bytes of the User Code field in the actual User Code block.

CC:0063.02.0D.11.007 The User Code Length MUST be in the range 4..10 if the User ID Status is different than 0x00 and 0xFE. The User Code Length MUST be set to 0 if the User ID Status is 0x00 or 0xFE.

User Code (N bytes)

This field is used to advertise the User Code currently set for the User ID in the actual User Code block.

CC:0063.02.0D.11.008 The length of this field in bytes MUST be according to the corresponding User Code Length field value. If the User Code Length is set to 0, this field MUST be omitted.

CC:0063.02.0D.11.009 Each byte in this field MUST be encoded with ASCII representation.

Next User Identifier (16 bits)

This field is used to specify the next User Identifier in use in the User Code database after the User Identifier advertised in the last User Code block.

CC:0063.02.0D.11.00A This field MUST be set to the next User Identifier having a User ID Status different than 0. The first byte MUST carry the most significant byte of the 16-bit value.

CC:0063.02.0D.11.00B The value 0 MUST indicate that the actual User Identifier is the last set at the sending node.

CC:0063.02.0D.11.00C If the status field is set to 0xFE for the last User Code block (User Identifier higher than the advertised supported users number in the Users Number Report Command), this field MUST be set to 0.

4.116.15 Master Code Set Command

This command is used to set the Master Code in the receiving node.

CC:0063.02.0E.11.001 This command MUST be ignored by a receiving node advertising no support for the Master Code functionality (MD Support) in the User Code Capabilities Report Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_USER_CODE (0x63)							
Command = MASTER_CODE_SET (0x0E)							
Reserved				Master Code Length			
Master Code 1							
…							
Master Code N							

Reserved

CC:0063.02.0E.11.002 This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Master Code Length (4 bits)

This field is used to advertise the length in bytes of the Master Code field in the command.

CC:0063.02.0E.11.003 This field MUST be set to 0 or in the range 4..10.

CC:0063.02.0E.11.004 Values in the range 4..10 MUST indicate that the receiving node MUST set the Master Code as indicated in the Master Code field.

CC:0063.02.0E.11.005 The Value 0 MUST indicate that the receiving node MUST deactivate the Master Code.

If this field is set to 0, this command MUST be ignored by a receiving node if it advertises no support for Master Code Deactivation (MCD) in the User Code Capabilities Report Command.

Master Code (N bytes)

This field is used to advertise the Master Code to be set for the node.

CC:0063.02.0E.11.006 The length of this field MUST be according to the Master Code Length field value.

Each byte in this field MUST be encoded with ASCII representation.

CC:0063.02.0E.11.007 A supporting node receiving a non-supported ASCII character MUST ignore the command.

A supporting node receiving a Master Code identical to one already set to another User ID MAY ignore the command.

When the Master Code is input, the node:

CC:0063.02.0E.11.008
- MUST indicate that the code is accepted

CC:0063.02.0E.11.009
- MUST grant access to administrator functionalities, such as the network settings and/or User Code management interface.

CC:0063.02.0E.11.00A Requirements in Table 146 (keypad modes) MUST take precedence over the above Master Code requirements

4.116.16 Master Code Get Command

This command is used to request the Master Code currently set at the receiving node.

CC:0063.02.0F.11.001 The Master Code Report Command MUST be returned in response to this command

CC:0063.02.0F.11.002 This command MUST NOT be issued via multicast addressing.

CC:0063.02.0F.11.003 A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_USER_CODE (0x63)							
Command = MASTER_CODE_GET (0x0F)							

4.116.17 Master Code Report Command

This command is used to advertise the Master Code currently set at the sending node.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_USER_CODE (0x63)							
Command = MASTER_CODE_REPORT (0x10)							
Reserved				Master Code Length			
Master Code 1							
…							
Master Code N							

Reserved

CC:0063.02.10.11.001 This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Master Code Length (4 bits)

This field is used to advertise the length in bytes of the Master Code field in the command.

CC:0063.02.10.11.002 This field MUST be in the range 4..10 if the Master Code is set and advertised in this command.

CC:0063.02.10.11.003 This field MUST be set to 0 if the Master Code is deactivated or if the sending node advertises no support for the Master Code functionality (MC Support) in the User Code Capabilities Report Command.

Master Code (N bytes)

This field is used to advertise the Master Code currently set at the sending node.

CC:0063.02.10.11.004 The length of this field MUST be according to the Master Code Length field value. If the Master Code Length is set to 0, this field MUST be omitted.

CC:0063.02.10.11.005 Each byte in this field MUST be encoded with ASCII representation.

4.116.18 User Code Checksum Get Command

This command is used to request a User Code checksum representing all the User Codes currently set at the receiving node.

CC:0063.02.11.11.001 This command MUST be ignored by a node advertising no support for the User Code Checksum functionality (UCC Support) in the User Code Capabilities Report Command.

CC:0063.02.11.11.002 The User Code Checksum Report Command MUST be returned in response to this command if the User Code Checksum functionality (UCC Support) is advertised as supported in the User Code Capabilities Report Command.

CC:0063.02.11.11.003 This command MUST NOT be issued via multicast addressing.

CC:0063.02.11.11.004 A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_USER_CODE (0x63)							
Command = USER_CODE_CHECKSUM_GET (0x11)							

4.116.19 User Code Checksum Report Command

This command is used to advertise the current User Code checksum representing all the User codes set at the sending node.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_USER_CODE (0x63)							
Command = USER_CODE_CHECKSUM_REPORT (0x12)							
User Code Checksum 1 (MSB)							
User Code Checksum 2 (LSB)							

User Code Checksum (16 bits)

This field is used to advertise the checksum representing all User Codes set at the sending node.

CC:0063.02.12.11.001 The checksum MUST be calculated using the CRC-CCITT polynomium using initialization value equal to 0x1D0F and 0x1021 (normal representation).

CC:0063.02.12.11.002 This field MUST be set to 0x0000 if no User Code is set at node.

CC:0063.02.12.11.003 The checksum data MUST be built by concatenating defined User Codes data (with User ID Status different than 0) in the ascending User ID order. The Master Code MUST NOT be part of the checksum calculation.

CC:0063.02.12.11.004 Each defined User Code data MUST be formatted as follows:

User ID (16 bits) | User ID Status (8 bits) | User Code (4..10 bytes)

For example, a node supporting 4 User IDs that are set as follow:

- User ID 1, status 4, code 9277
- User ID 2, status 0
- User ID 3, status 1, code 88473
- User ID 4, status 0

CC:0063.02.12.11.005 In this case, User Code data for User ID 1 and 3 MUST be concatenated to obtain the checksum data:

0x0001 | 0x04 | 0x39 32 37 37 | 0x0003 | 0x01 | 0x38 38 34 37 33

CC:0063.02.12.11.006 The checksum data MUST be: 0x000104393237370003013838343733.

The returned User Code Checksum field MUST be set to: 0xEAAD.

4.117 Window Covering Command Class, version 1

The Window Covering Command Class is used to control window covering devices.
The Window Covering Command Class is an actuator control command class. Refer to 3.6.

4.117.1 Terminology

A window covering device may be closed or open. The term closed represents the lowest light throughput, while open represents the highest light throughput.

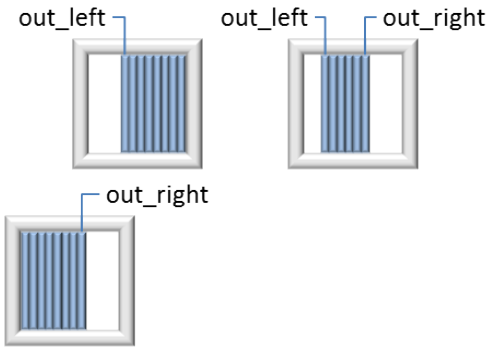
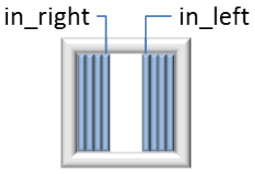
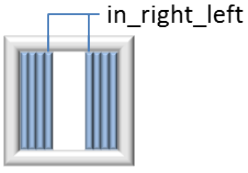
A window covering device may provide one or more properties such as up/down movement combined with for example slats angle control.

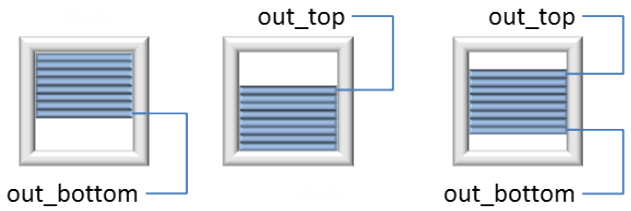
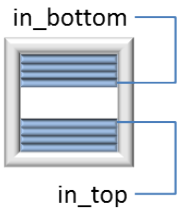
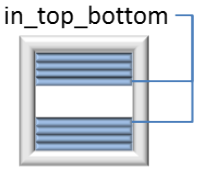
Each of the available properties is advertised. A window covering device may offer precise control of the position of a property or it may offer a more limited movement control, where it is possible to start and stop movement in one of two directions but where the target position cannot be specified.

If a window covering parameter is marked as position unknown, this indicates that the actual property can only be controlled via start and stop of movement.

The parameters controlling edges are identified as outlined in Table 149.

Table 149, Identification of edges

Edge Title	Description	Usage examples
out_left	Outbound edge towards the left	
out_right	Outbound edge towards the right	
in_left	Inbound edge towards the left	
in_right	Inbound edge towards the right	
in_right_left	Inbound edges controlled horizontally as one	

Edge Title	Description	Usage examples
out_bottom	Outbound edge towards the bottom	
out_top	Outbound edge towards the top	
in_bottom	Inbound edge towards the bottom	
in_top	Inbound edge towards the top	
in_top_bottom	Inbound edges controlled vertically as one	

4.117.2 Compatibility considerations

This Command Class replaces the following command classes:

- Basic Window Covering Command Class [OBSOLETED]
- Move To Position Window Covering Command Class [OBSOLETED]

4.117.2.1 Motor Control Device Class support

A node supporting this Command Class SHOULD comply implement the Motor Control Specific Device Class B or C (refer to [1], [10] or [22]) for backwards compatibility with existing window covering control applications.

4.117.2.2 Command Class dependencies

A node supporting the Window Covering Command Class, Version 1 MUST support the Multilevel Switch Command Class, version 3 or newer.

With the exception for slats angle control, the following mapping MUST apply:

- The Multilevel Switch value 0x00 MUST represent the least light, i.e. covering fully closed.
- The Multilevel Switch value 0x63 MUST represent the most light, i.e. covering fully opened.

For slats angle control, the following mapping MUST apply:

- The Multilevel Switch value 0x00 MUST represent slats closed to the one side.
- The Multilevel Switch value 0x32 MUST represent slats open.
- The Multilevel Switch value 0x63 MUST represent slats closed to the other side.

By mapping the Multilevel Switch value 0 to the closed position, a multicasted or broadcasted

Multilevel Switch Command that turns off light sources will also cause window coverings to close, thus reducing light from the outside.

A supporting node MAY use the Multilevel Switch Command Class to control multiple features like up/down control as well as slats angle control. For instance, the Start Level Change command may start changing the slats angle towards the extreme angle and when reaching that extreme angle, it may start moving up or down.
This way, multiple functions can be controlled from a general purpose remote control, although with less convenience than if a dedicated window covering remote control is used.

4.117.3 Window Covering Parameters

Each distinct property of a window covering node is accessed via a dedicated parameter.

Two variants are defined for many parameters: One allows full position control, including movement control. The other variant is limited to movement control.

If a node supports full position control of a property (even parameter IDs), it MUST NOT support the corresonding parameter limited to movement control (odd parameter IDs).

The term “Closed” represents the lowest light throughput, while “Open” represents the highest light throughput of the window covering.

Table 150, Window Covering Parameter IDs

Parm ID	Description	Value range	Encoding
Outbound horizontal control			
0	out_left: Outbound edge towards the left Right/Left movement (Position unknown)	NA	Level Change Up = Opening Level Change Down = Closing
1	out_left: Outbound edge towards the left Right/Left position Parm 0 MUST NOT be supported	0x00 – 0x63	0x00 = Closed 0x63 = Open
2	out_right: Outbound edge towards the right Right/Left movement (Position unknown)	NA	Level Change Up = Opening Level Change Down = Closing
3	out_right: Outbound edge towards the right Right/Left position Parm 2 MUST NOT be supported	0x00 – 0x63	0x00 = Closed 0x63 = Open

Parm ID	Description	Value range	Encoding
Inbound horizontal control			
4	in_left: Inbound edge towards the left Right/Left movement (Position unknown)	NA	Level Change Down = Closing Level Change Up = Opening
5	in_left: Inbound edge towards the left Right/Left position Parm 4 MUST NOT be supported	0x00 – 0x63	0x00 = Closed 0x63 = Open
6	in_right: Inbound edge towards the right Right/Left movement (Position unknown)	NA	Level Change Down = Closing Level Change Up = Opening
7	in_right: Inbound edge towards the right Right/Left position Parm 6 MUST NOT be supported	0x00 – 0x63	0x00 = Closed 0x63 = Open
8	in_right_left: Inbound edges controlled horizontally as one Right/Left movement (Position unknown)	NA	Level Change Down = Closing Level Change Up = Opening
9	in_right_left: Inbound edges controlled horizontally as one Right/Left position Parm 8 MUST NOT be supported	0x00 – 0x63	0x00 = Closed 0x63 = Open
Angle control of vertical slats			
10	Vertical slats angle Right/Left movement (Position unknown)	NA	Level Change Down = Closing; to the right inside Level Change Up = Closing; to the left inside (Open is passed midway between the two closing positions)
11	Vertical slats angle Right/Left position Parm 10 MUST NOT be supported	0x00 – 0x63	0x00 = Closed; to the right inside 0x32 = Open 0x63 = Closed; to the left inside

Parm ID	Description	Value range	Encoding
Outbound vertical control			
12	out_bottom: Outbound edge towards the bottom Up/Down movement (Position unknown)	NA	Level Change Down = Closing Level Change Up = Opening
13	out_bottom: Outbound edge towards the bottom Up/Down position Parm 12 MUST NOT be supported	0x00 – 0x63	0x00 = Closed 0x63 = Open
14	out_top: Outbound edge towards the top Up/Down movement (Position unknown)	NA	Level Change Down = Closing Level Change Up = Opening
15	out_top: Outbound edge towards the top Up/Down position Parm 14 MUST NOT be supported	0x00 – 0x63	0x00 = Closed 0x63 = Open
Inbound vertical control			
16	in_bottom: Inbound edge towards the bottom Up/Down movement (Position unknown)	NA	Level Change Down = Closing Level Change Up = Opening
17	in_bottom: Inbound edge towards the bottom Up/Down position Parm 16 MUST NOT be supported	0x00 – 0x63	0x00 = Closed 0x63 = Open
18	in_top: Inbound edge towards the top Up/Down movement (Position unknown)	NA	Level Change Down = Closing Level Change Up = Opening
19	in_top: Inbound edge towards the top Up/Down position Parm 18 MUST NOT be supported	0x00 – 0x63	0x00 = Closed 0x63 = Open

Parm ID	Description	Value range	Encoding
20	in_top_bottom: Inbound edges controlled vertically as one Up/Down movement (Position unknown)	NA	Level Change Down = Closing Level Change Up = Opening
21	in_top_bottom: Inbound edges controlled vertically as one Up/Down position Parm 20 MUST NOT be supported	0x00 – 0x63	0x00 = Closed 0x63 = Open
Angle control of horizontal slats			
22	Horizontal slats angle Up/Down movement (Position unknown)	NA	Level Change Down = Closing; up inside Level Change Up = Closing; down inside (Open is passed midway between the two closing positions)
23	Horizontal slats angle Up/Down position Parm 22 MUST NOT be supported	0x00 – 0x63	0x00 = Closed; up inside 0x32 = Open 0x63 = Closed; down inside

All other values are reserved and MUST NOT be used by a sending node. Reserved values MUST be ignored by a receiving node.

A device MAY implement up to 100 hardware levels (including 0) for a given parameter. If a device implements less than 100 hardware levels, the hardware levels MUST be distributed uniformly over the entire range.
The mapping of parameter values to hardware levels MUST be monotonous, i.e. a higher value MUST be mapped to either the same or a higher hardware level. An example is found in Table 151.

Table 151, Parameter value mapping to a limited number of hardware levels (example)

Horizontal Slats Angle	Hardware level
0x00..0x13	100% Up inside
0x14..0x27	50% Up inside
0x28..0x3B	Open
0x3C..0x4F	50% Down inside
0x50..0x63	100% Down inside

4.117.4 Window Covering Supported Get Command

This command is used to request the supported properties of a device.

The Window Covering Supported Report Command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_WINDOW_COVERING (0x6A)							
Command = WINDOW_COVERING_SUPPORTED_GET (0x01)							

4.117.5 Window Covering Supported Report Command

This command is used to advertise the supported properties of a device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_WINDOW_COVERING (0x6A)							
Command = WINDOW_COVERING_SUPPORTED_REPORT (0x02)							
Reserved				Number of Parameter Mask bytes			
Parameter Mask 1							
…							
Parameter Mask N							

Reserved

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Number of Parameter Mask bytes (4 bits)

The Number of Parameter Masks field MUST advertise the number of bytes carrying the Covering Parameter Mask field.

The value MUST be in the range 1..15.

Parameter Mask (N Bytes)

The Parameter Mask field MUST advertise the Parameters supported by the device.

The length of this field MUST be advertised by the Number of Parameter Masks field.

- Bit 0 in Bit Mask 1 indicates if Parameter ID 0 is supported
- Bit 1 in Bit Mask 1 indicates if Parameter ID 1 is supported
- …

For the definition of Parameter IDs, refer to Table 150.

4.117.6 Window Covering Get Command

This command is used to request the status of a specified Covering Parameter.

The Window Covering Report command MUST be returned in response to this command.

This command MUST NOT be issued via multicast addressing.
A receiving node MUST NOT return a response if this command is received via multicast addressing. The Z-Wave Multicast frame, the broadcast NodeID and the Multi Channel multi-End Point destination are all considered multicast addressing methods.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_WINDOW_COVERING (0x6A)							
Command = WINDOW_COVERING_GET (0x03)							
Parameter ID							

Parameter ID (8 bits)

This field MUST specify the Parameter for which the status is requested.
For the definition of Parameter IDs, refer to Table 150.

A node receiving an even Parameter ID (marked as “position unknown”) MUST set the *Current Value* and *Target Value* fields to 0x00 and the *Duration* field to 0xFE in the returned Report if the window covering is not moving (i.e., stationary).

4.117.7 Window Covering Report Command

This command is used to advertise the status of a Parameter.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_WINDOW_COVERING (0x6A)							
Command = WINDOW_COVERING_REPORT (0x04)							
Parameter ID							
Current Value							
Target Value							
Duration							

Parameter ID (8 bits)

This field MUST advertise the Parameter covered by this report.
For the definition of Parameter IDs, refer to Table 150.

Current Value (8 bits)

The Current Value field MUST advertise the current value of the Parameter identified by the Parameter ID field.

The Current Value SHOULD be identical to the Target Value when a transition has ended.
For the definition of valid parameter values, refer to Table 150.

Target Value (8 bits)

The Target Value field MUST advertise the target value of an ongoing transition or the most recent transition for the advertised Parameter ID.

If a transition is initiated in an interactive fashion via a local user interface or via a Start Level Change command, the advertised Target Value MUST be 0x00 or 0x63, depending on the direction.

For the definition of valid parameter values, refer to Table 150.

Duration (8 bits)

The Duration field SHOULD advertise the time needed to reach the Target Value at the actual transition rate. The encoding of the Duration field MUST be according to Table 8, Actuator Command Class Duration Report encoding.

The Duration is defined as the interval from start of the transition until the Target Value is reached.

4.117.8 Window Covering Set Command

This command is used to control one or more parameters in a window covering device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_WINDOW_COVERING (0x6A)							
Command = WINDOW_COVERING_SET (0x05)							
Reserved			Parameter Count				
Parameter ID 1							
Value 1							
...							
Parameter ID N							
Value N							
Duration							

Reserved

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Parameter Count (5 bits)

This field MUST specify the number of (Parameter ID, Value) datasets contained in the Window Covering Set command.

Parameter ID (N bytes)

This field MUST specify the Parameter to receive a new value.

A receiving node MUST ignore even Parameters ID (marked as “position unknown”).
For the definition of Parameter IDs, refer to Table 150.

Value (N bytes)

This field MUST specify the value of the Parameter identified by the Parameter ID field.
For the definition of valid parameter ID values, refer to Table 150.

Duration (8 bits)

The Duration field MUST specify the time that the transition should take from the current value to the new target value.
A supporting device SHOULD respect the specified Duration value.

The encoding of the Duration field MUST be according to Table 7, Actuator Command Class Duration Set encoding.

4.117.9 Window Covering Start Level Change Command

This command is used to initiate a transition of one parameter to a new level.

A receiving node MUST initiate the transition to a new value for the specified Parameter ID.

<table>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="8">Command Class = COMMAND_CLASS_WINDOW_COVERING (0x6A)</td></tr>
<tr><td colspan="8">Command = WINDOW_COVERING_START_LEVEL_CHANGE (0x06)</td></tr>
<tr><td>Res</td><td>Up/
Down</td><td colspan="6">Res</td></tr>
<tr><td colspan="8">Parameter ID</td></tr>
<tr><td colspan="8">Duration</td></tr>
</table>

Up/Down (1 bit)

This field MUST specify the direction of the level change.

If the Up/Down bit is set to 0 the level change MUST be increasing.
If the Up/Down bit is set to 1 the level change MUST be decreasing.

Reserved

This field MUST be set to 0 by a sending node and MUST be ignored by a receiving node.

Parameter ID (8 bits)

This field MUST specify the Parameter to start a transition.

A node receiving this command MUST accept all its supported parameter IDs (even and odd).
For the definition of Parameter IDs, refer to Table 150.

Duration (8 bits)

The level change rate MUST be calculated to match a level change from the minimum value to the maximum value during the time specified by the Duration field.
A supporting device SHOULD respect the specified Duration value.

The encoding of the Duration field MUST be according to Table 7, Actuator Command Class Duration Set encoding.

4.117.10 Window Covering Stop Level Change Command

This command is used to stop an ongoing transition.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_WINDOW_COVERING (0x6A)							
Command = WINDOW_COVERING_STOP_LEVEL_CHANGE (0x07)							
Parameter ID							

Parameter ID (8 bits)

This field MUST specify the parameter to stop a transition.

A receiving node MUST stop the transition if the specified Parameter ID is currently in transition to a new value.
A receiving node MUST NOT stop ongoing transitions for other Parameter IDs than the one specified.

A node receiving this command MUST accept all parameter IDs supported by the device.
For the definition of Parameter IDs, refer to Table 150.

APPENDIX AASCII 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:

Table 152, The standard ASCII Table

	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	{	|	}	~	

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:

Table 153, OEM Extended ASCII Table

	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	ß	Γ	π	Σ	σ	µ	τ	Φ	Θ	Ω	δ	∞	φ	∈	∩
F	≡	±	≥	≤	⌠	⌡	÷	≈	°	∙	•	√	ⁿ	²	■	

Below are listed codes for players, radios etc. as an alternative to the OEM Extended ASCII codes. Undefined values MUST be ignored.

Table 154, Players Table

	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

References

[1] Silicon Labs, SDS10242, Software Design Spec., Z-Wave Device Class Specification.
[2] Silicon Labs, SDS12657, Z-Wave Command Class Specification A-M.
[3] Silicon Labs, SDS12652, Z-Wave Command Class Specification N-Z.
[4] IETF RFC 4861, Neighbor Discovery for IP version 6 (IPv6), http://tools.ietf.org/pdf/rfc4861.pdf
[5] IETF RFC 3122, Extensions to IPv6 Neighbor Discovery for Inverse Discovery Specification, http://tools.ietf.org/pdf/rfc3122.pdf
[6] IETF RFC 2119, Key words for use in RFCs to Indicate Requirement Levels, http://tools.ietf.org/pdf/rfc2119.pdf
[7] IETF RFC 2460, Internet Protocol, Version 6 (IPv6) Specification, http://tools.ietf.org/pdf/rfc2460.pdf
[8] IETF RFC 4291, IP Version6 Addressing Architecture, http://tools.ietf.org/pdf/rfc4291.pdf
[9] Silicon Labs, SDS11846, Z-Wave Plus Role Types Specification.
[10] Silicon Labs, SDS11847, Z-Wave Plus Device Type Specification.
[11] Graphical UI elements, http://en.wikipedia.org/wiki/Graphical_user_interface_elements
[12] Silicon Labs, SDS13425. Z-Wave Plus Assigned Manufacturer IDs.
[13] Silicon Labs, SDS13548, List of defined Z-Wave Command Classes
[14] Silicon Labs, SDS13783, Z-Wave Transport-Encapsulation Command Class Specification
[15] Silicon Labs, SDS13782, Z-Wave Management Command Class Specification
[16] Silicon Labs, SDS13784, Z-Wave Network-Protocol Command Class Specification
[17] Silicon Labs, SDS13812, Multilevel Sensor Command Class, list of assigned Multilevel Sensor types and scales
[18] Silicon Labs, SDS13713, Notification Command Class, list of assigned Notifications
[19] Silicon Labs, SDS13907, Simple AV Command Class, list of assigned AV Control codes
[20] Silicon Labs, SDS14223, Z-Wave Command Class Control Specification
[21] Silicon Labs, SDS14305, Meter Table Monitor Command Class, list of assigned types, scales and datasets
[22] Silicon Labs, SDS14224, Z-Wave Plus v2 Device Type Specification.
[23] Silicon Labs, SDS14223, Z-Wave Command Class Control Specification
[24]