ZIGBEE SMART ENERGY STANDARD

ZigBee Smart Energy Standard

ZigBee Profile: 0x0109

Revision 19

Version 1.2a

ZigBee Document:	07-5356-19
Date Approved:	December 3, 2014
Sponsored by:	ZigBee Alliance
Accepted by:	ZigBee Alliance Board of Directors.
Abstract Keywords:	ZigBee, Profile, AMI, Application Framework, Smart Energy, Standard.

December 3, 2014

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

This page intentionally blank

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Notice of Use and Disclosure

Copyright © ZigBee Alliance, Inc. (2007-2014). All rights Reserved. The information within this document is the property of the ZigBee Alliance and its use and disclosure are restricted.

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

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

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

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

Table of Contents

Notice of Use and Disclosure 3
Table of Contents 4
List of Tables 18
List of Figures 24
Participants 30
Document History 31
1 Introduction 42
1.1 Scope 42
1.2 Purpose 42
1.3 Provisional Features 42
2 References 43
2.1 References 43
2.1.1 ZigBee Alliance Documents 43
2.1.2 External Reference Documents 43
3 Definitions 45
3.1 Conformance Levels 45
3.2 ZigBee Definitions 45
3.3 Smart Energy Definitions 46
4 Acronyms And Abbreviations 48
5 Profile Description 50
5.1 A ZigBee Smart Energy Network 50
5.2 ZigBee Stack Profile 53
5.2.1 MAC Data Polling (NLME_Requests) 54
5.2.2 Application Level Queries 54
5.2.3 ZigBee Coordinator and Trust Center Recommendations 55
5.3 Startup Attribute Set (SAS) 55
5.3.1 Startup Parameters 55
5.3.2 Join Parameters 57
5.3.3 Security Parameters 58
5.3.4 End Device Parameters 59
5.3.5 Link Status Parameters 59
5.3.6 Concentrator Parameters 59
5.3.7 APS Transport Parameters 59
5.3.8 APS Fragmentation Parameters 60
5.3.9 Binding Parameters 60
5.4 Smart Energy Profile Security 61
5.4.1 Joining with Preinstalled Trust Center Link Keys 61
5.4.2 Re-Joining a Secured Network 63
5.4.3 Devices Leaving the Network 70
5.4.4 Updating the Network Key 70

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

5.4.5 Updating the Link Key 71
5.4.6 Cluster Usage of Security Keys 73
5.4.7 Key Establishment Related Security Policies 75
5.4.8 Security Best Practices 78
5.5 Commissioning 81
5.5.1 Forming the Network (Start-up Sequence) 82
5.5.2 Support for Commissioning Modes 82
5.5.3 Commissioning Documentation Best Practices 82
5.5.4 Commissioning Procedure for Different Network Types 83
5.5.5 ZigBee Smart Energy Joining, Service Discovery, and Device Binding Requirements 84
5.6 Federated Trust Center Application Guidelines 90
5.7 Multiple ESI Application Guidelines 91
5.7.1 Overview 91
5.7.2 Device Behavior 91
5.8 Other Smart Energy Profile Requirements and Best Practices 94
5.8.1 Preferred Channel Usage 94
5.8.2 Broadcast Policy 94
5.8.3 Frequency Agility 94
5.8.4 Key Updates 94
5.9 Coexistence and Interoperability with HA Devices 95
5.10 Device Descriptions 95
5.11 ZigBee Cluster Library (ZCL) 96
5.12 Cluster List and IDs 97
5.12.1 ZCL General Clusters 98
5.13 Coexistence with devices using other Profiles 99
5.13.1 Requirements for the Smart Energy Trust Center 100
5.13.2 Requirements for Smart Energy Devices 100
5.13.3 Requirements for Devices from other Profiles 101
6 Device Specifications 102
6.1 Common Clusters 102
6.1.1 Optional Support for Clusters with Reporting Capability 102
6.1.2 Manufacturer-Specific Clusters 102
6.1.3 Cluster Usage Restrictions 102
6.1.4 Identify Cluster Best Practices 103
6.1.5 Inter-PAN Communication 103
6.2 Feature and Function Description 103
6.3 Smart Energy Devices 105
6.3.1 Energy Service Interface 105
6.3.2 Metering Device 106
6.3.3 In-Home Display Device 108
6.3.4 Programmable Communicating Thermostat (PCT) Device 109
6.3.5 Load Control Device 109

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

6.3.6 Range Extender Device 110
6.3.7 Smart Appliance Device 110
6.3.8 Prepayment Terminal Device 111
6.3.9 Physical Device 112
6.3.10 Remote Communications Device 113
Annex A Candidate ZCL Material For Use with this Profile 115
A.1 New Status Enumeration 115
A.2 New Attribute Reporting Status Indication 115
A.2.1 *Attribute Reporting Status* Attribute 115
Annex B Enhanced Inter-PAN Transmission Mechanism 117
B.1 Scope and Purpose 117
B.2 General Description 117
B.2.1 What Enhanced Inter-PAN Transmission Does 117
B.3 Service Specification 118
B.3.1 The INTRP-DATA.request Primitive 119
B.3.1.1 Semantics of the Service Primitive 119
B.3.1.2 When Generated 121
B.3.1.3 Effect on Receipt 121
B.3.2 The INTRP-DATA.confirm Primitive 121
B.3.2.1 Semantics of the Service Primitive 121
B.3.2.2 When Generated 122
B.3.2.3 Effect on Receipt 122
B.3.3 The INTRP-DATA.indication Primitive 122
B.3.3.1 Semantics of the Service Primitive 122
B.3.3.2 When Generated 124
B.3.3.3 Effect on Receipt 124
B.3.4 Qualifying and Testing of Inter-PAN Messages 124
B.4 Frame Formats 124
B.5 Frame Processing 127
B.5.1 Enhanced Inter-PAN Transmission 127
B.5.2 Enhanced Inter-PAN Reception 128
B.6 Initiating a Enhanced Inter-PAN Interaction 129
B.7 Best Practices 130
B.8 Security Requirements 130
Annex C Key Establishment Cluster 131
C.1 Scope and Purpose 131
C.2 General Description 131
C.2.1 Introduction 131
C.2.2 Security Credentials 131
C.2.3 Network Security 131
C.2.4 Key Establishment 131
C.2.5 Symmetric Key Key Establishment 132

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

C.2.6 Public Key Key Establishment 133
C.2.7 General Exchange 133
C.2.7.1 Exchange Static and Ephemeral Data 134
C.2.7.2 Generate Key Bitstream 135
C.2.7.3 Derive MAC Key and Key Data 135
C.2.7.4 Confirm Key Using MAC 135
C.3 Cluster List 136
C.3.1 Key Establishment Cluster 136
C.3.1.1 Overview 136
C.3.1.1.1 Negotiating the Key Establishment Suite 138
C.3.1.2 Server 138
C.3.1.2.1 Dependencies 138
C.3.1.2.2 Attributes 138
C.3.1.2.3 Commands Received 139
C.3.1.2.4 Commands Generated 144
C.3.1.3 Client 145
C.3.1.3.1 Dependencies 145
C.3.1.3.2 Attributes 145
C.3.1.3.3 Commands Received 146
C.3.1.3.4 Commands Generated 151
C.4 Application Implementation 151
C.4.1 Network Security for Smart Energy Networks 151
C.4.2 Certificate-Based Key Establishment 152
C.4.2.1 Notation and Representation 153
C.4.2.1.1 Strings and String Operations 153
C.4.2.1.2 Integers and their Representation 153
C.4.2.1.3 Entities 153
C.4.2.2 Cryptographic Suite 1 Building Blocks 153
C.4.2.2.1 Elliptic-Curve Domain Parameters 153
C.4.2.2.2 Elliptic-Curve Point Representation 153
C.4.2.2.3 Elliptic-Curve Key Pair 154
C.4.2.2.4 ECC Implicit Certificates 154
C.4.2.2.5 Block-Cipher 155
C.4.2.2.6 Cryptographic Hash Function 155
C.4.2.2.7 Keyed Hash Function for Message Authentication 155
C.4.2.2.8 Derived Shared Secret 155
C.4.2.3 Cryptographic Suite 2 Building Blocks 155
C.4.2.3.1 Elliptic-Curve Point Representation 156
C.4.2.3.2 Elliptic-Curve Key Pair 156
C.4.2.3.3 ECC Implicit Certificates 156
C.4.2.3.4 Block-Cipher 158
C.4.2.3.5 Cryptographic Hash Function 158
C.4.2.3.6 Keyed Hash Function for Message Authentication 158

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

257 C.4.2.3.7 Derived Shared Secret 158
258 C.4.2.4 Certificate-Based Key-Establishment 158
259 C.4.2.4.1 Exchange Ephemeral Data 160
260 C.4.2.4.2 Validate Implicit Certificates 160
261 C.4.2.4.3 Derive Keying Material 160
262 C.4.2.4.4 Confirm Keys 162
263 C.5 Key Establishment Test Vectors for Cryptographic Suite 1 162
264 C.5.1 Preconfigured Data 162
265 C.5.1.1 CA Public Key 163
266 C.5.1.2 Responder Data 163
267 C.5.1.3 Initiator Data 163
268 C.5.2 Key Establishment Messages 164
269 C.5.2.1 Initiate Key Establishment Request 165
270 C.5.2.2 Initiate Key Establishment Response 166
271 C.5.2.3 Ephemeral Data Request 167
272 C.5.2.4 Ephemeral Data Response 168
273 C.5.2.5 Confirm Key Request 168
274 C.5.2.6 Confirm Key Response 169
275 C.5.3 Data Transformation 170
276 C.5.3.1 ECMQV Primitives 170
277 C.5.3.2 Key Derivation Function (KDF) 171
278 C.5.3.3 Initiator Transform 171
279 C.5.3.3.1 Ephemeral Data 171
280 C.5.3.3.2 Step Summary 171
281 C.5.3.3.3 Detailed Steps 171
282 C.5.3.4 Responder Transform 173
283 C.5.3.4.1 Ephemeral Data 173
284 C.5.3.4.2 Step Summary 173
285 C.5.3.4.3 Detailed Steps 173
286 C.6 Key Establishment Test Vectors for Cryptographic Suite 2 175
287 C.6.1 Preconfigured Data 175
288 C.6.1.1 CA Public Key 175
289 C.6.1.2 Responder Data 175
290 C.6.1.3 Initiator Data 176
291 C.6.2 Key Establishment Messages 177
292 C.6.2.1 Initiate Key Establishment Request 177
293 C.6.2.2 Initiate Key Establishment Response 178
294 C.6.2.3 Ephemeral Data Request 179
295 C.6.2.4 Ephemeral Data Response 179
296 C.6.2.5 Confirm Key Request 180
297 C.6.2.6 Confirm Key Response 181
298 C.6.3 Data Transformation 181

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

C.6.3.1 ECMQV Primitives 182
C.6.3.2 Key Derivation Function (KDF) 182
C.6.3.3 Initiator Transform 182
C.6.3.3.1 Ephemeral Data 182
C.6.3.3.2 Step Summary 183
C.6.3.3.3 Detailed Steps 183
C.6.3.4 Responder Transform 185
C.6.3.4.1 Ephemeral Data 185
C.6.3.4.2 Step Summary 185
C.6.3.4.3 Detailed Steps 185
Annex D Smart Energy Cluster Descriptions 188
D.1 Annex Guidelines 188
D.1.1 Client/Server Model Information 188
D.1.2 Interpretation of Reserved Field Values or Bitmaps 188
D.2 Demand Response and Load Control Cluster 189
D.2.1 Overview 189
D.2.2 Server 189
D.2.2.1 Dependencies 189
D.2.2.2 Attributes 190
D.2.2.3 Commands Generated 190
D.2.2.3.1 Load Control Event Command 190
D.2.2.3.2 Cancel Load Control Event Command 196
D.2.2.3.3 Cancel All Load Control Events Command 198
D.2.2.4 Commands Received 198
D.2.3 Client 199
D.2.3.1 Dependencies 199
D.2.3.2 Client Cluster Attributes 199
D.2.3.2.1 Utility Enrollment Group Attribute 200
D.2.3.2.2 Start Randomization Minutes Attribute 200
D.2.3.2.3 End Randomization Minutes Attribute 200
D.2.3.2.4 DeviceClassValue Attribute 200
D.2.3.3 Commands Generated 201
D.2.3.3.1 Report Event Status Command 201
D.2.3.3.2 Get Scheduled Events Command 204
D.2.3.4 Commands Received 205
D.2.3.5 Attribute Reporting 205
D.2.4 Application Guidelines 205
D.2.4.1 Load Control Rules, Server 206
D.2.4.1.1 Load Control Server, Identifying Use of SetPoint and Offset Fields 206
D.2.4.1.2 Load Control Server, Editing of Scheduled Events 207
D.2.4.2 Load Control Rules, Client 207
D.2.4.2.1 Start and Stop Randomization 207
D.2.4.2.2 Editing of DR Control Parameters 207

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

342 D.2.4.2.3 Response to Price Events + Load Control Events 207
343 D.2.4.2.4 Opt-Out Messages 207
344 D.2.4.2.5 Thermostat/HVAC Controls 207
345 D.2.4.2.6 Demand Response and Load Control Transaction Examples 208
346 D.3 Metering Cluster 211
347 D.3.1 Overview 211
348 D.3.2 Server 214
349 D.3.2.1 Dependencies 214
350 D.3.2.2 Attributes 214
351 D.3.2.2.1 Reading Information Set 214
352 D.3.2.2.2 Summation TOU Information Set 224
353 D.3.2.2.3 Meter Status Attribute Set 228
354 D.3.2.2.4 Formatting 235
355 D.3.2.2.5 Historical Consumption Attribute 248
356 D.3.2.2.6 Load Profile Configuration 258
357 D.3.2.2.7 Supply Limit Attributes 259
358 D.3.2.2.8 Block Information Attribute Set (Delivered) 261
359 D.3.2.2.9 Alarms Set 266
360 D.3.2.2.10 Block Information Attribute Set (Received) 272
361 D.3.2.2.11 Meter Billing Attribute Set 276
362 D.3.2.2.12 Supply Control Attribute Set 278
363 D.3.2.2.13 Alternative Historical Consumption Attribute Set 279
364 D.3.2.3 Server Commands 288
365 D.3.2.3.1 Commands Generated 288
366 D.3.3 Client 316
367 D.3.3.1 Dependencies 316
368 D.3.3.2 Attributes 316
369 D.3.3.2.1 Notification Attribute Set 317
370 D.3.3.3 Client Commands 322
371 D.3.3.3.1 Commands Generated 322
372 D.3.4 Metering Application Guidelines 338
373 D.3.4.1 Attribute Reporting 338
374 D.3.4.2 Fast Polling or Reporting for Monitoring Energy Savings 338
375 D.3.4.3 Metering Data Updates 339
376 D.3.4.3.1 Fast Polling Periods 339
377 D.3.4.4 Mirroring 339
378 D.3.4.4.1 Discovery 339
379 D.3.4.4.2 Mirror Attributes 339
380 D.3.4.4.3 Two Way Mirror for BOMD 340
381 D.3.4.5 An Introduction to Snapshots 348
382 D.3.4.6 Supply Control 348
383 D.4 Price Cluster 350

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

D.4.1 Overview 350
D.4.2 Server 350
D.4.2.1 Dependencies 350
D.4.2.2 Attributes 350
D.4.2.2.1 Tier Label (Delivered) Set 351
D.4.2.2.2 Block Threshold (Delivered) Set 353
D.4.2.2.3 Block Period (Delivered) Set 356
D.4.2.2.4 Commodity Set 357
D.4.2.2.5 Block Price Information (Delivered) Set 360
D.4.2.2.6 Extended Price Information (Delivered) Set 364
D.4.2.2.7 Tariff Information (Delivered) Attribute Set 365
D.4.2.2.8 Billing Information (Delivered) Attribute Set 369
D.4.2.2.9 Credit Payment Attribute Set 370
D.4.2.2.10 Received Tier Label Attribute Set 372
D.4.2.2.11 Received Block Threshold Attribute Set 373
D.4.2.2.12 Received Block Period Attribute Set 374
D.4.2.2.13 Received Block Price Information Attribute Set 374
D.4.2.2.14 Received Extended Price Information Attribute Set 375
D.4.2.2.15 Received Tariff Information Attribute Set 376
D.4.2.2.16 Received Billing Information Attribute Set 377
D.4.2.3 Commands Received 378
D.4.2.3.1 Error Handling 379
D.4.2.3.2 GetCurrentPrice Command 379
D.4.2.3.3 GetScheduledPrices Command 380
D.4.2.3.4 PriceAcknowledgement Command 381
D.4.2.3.5 GetBlockPeriod(s) Command 382
D.4.2.3.6 GetConversionFactor Command 383
D.4.2.3.7 GetCalorificValue Command 384
D.4.2.3.8 GetTariffInformation Command 385
D.4.2.3.9 GetPriceMatrix Command 386
D.4.2.3.10 GetBlockThresholds Command 387
D.4.2.3.11 GetCO_2Value Command 387
D.4.2.3.12 GetTierLabels Command 388
D.4.2.3.13 GetBillingPeriod Command 389
D.4.2.3.14 GetConsolidatedBill Command 390
D.4.2.3.15 CPPEventResponse Command 391
D.4.2.3.16 GetCreditPayment Command 392
D.4.2.3.17 GetCurrencyConversion Command 392
D.4.2.3.18 GetTariffCancellation Command 393
D.4.2.4 Commands Generated 393
D.4.2.4.1 Publish Price Command 394
D.4.2.4.2 Publish Block Period Command 405
D.4.2.4.3 PublishConversionFactor Command 409

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

D.4.2.4.4 PublishCalorificValue Command 410
D.4.2.4.5 PublishTariffInformation Command 410
D.4.2.4.6 PublishPriceMatrix Command 413
D.4.2.4.7 PublishBlockThresholds Command 416
D.4.2.4.8 PublishCO_2Value Command 418
D.4.2.4.9 PublishTierLabels Command 420
D.4.2.4.10 PublishBillingPeriod Command 421
D.4.2.4.11 PublishConsolidatedBill Command 422
D.4.2.4.12 PublishCPPEvent Command 424
D.4.2.4.13 PublishCreditPayment Command 427
D.4.2.4.14 PublishCurrencyConversion Command 428
D.4.2.4.15 CancelTariff Command 429
D.4.3 Client 430
D.4.3.1 Dependencies 430
D.4.3.2 Attributes 431
D.4.3.2.1 PriceIncreaseRandomizeMinutes Attribute 431
D.4.3.2.2 PriceDecreaseRandomizeMinutes Attribute 432
D.4.3.2.3 CommodityType Attribute 432
D.4.3.3 Commands Received 432
D.4.3.4 Commands Generated 432
D.4.4 Application Guidelines 432
D.4.4.1 Registering for Commands 432
D.4.4.2 Attribute Reporting 432
D.4.4.3 Block Tariffs 433
D.4.4.3.1 TOU Charging Only 433
D.4.4.3.2 Block Charging only 433
D.4.4.3.3 Block/TOU Combination Charging 434
D.4.4.3.4 Application Guidelines for Block Pricing under specific events 434
D.4.4.4 Handling of Enhanced Tariffs 435
D.4.4.4.1 Block Charging 435
D.4.4.4.2 TOU Charging 436
D.4.4.4.3 TOU/Block Charging 436
D.4.4.4.4 Critical Peak Pricing 436
D.4.4.4.5 Generation Charging 437
D.5 Messaging Cluster 438
D.5.1 Overview 438
D.5.2 Server 438
D.5.2.1 Dependencies 438
D.5.2.2 Attributes 438
D.5.2.3 Commands Generated 438
D.5.2.3.1 Display Message Command 439
D.5.2.3.2 Cancel Message Command 442

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

D.5.2.3.3 Display Protected Message Command 442
D.5.2.3.4 Cancel All Messages Command 443
D.5.3 Client 443
D.5.3.1 Dependencies 443
D.5.3.2 Attributes 443
D.5.3.3 Commands Generated 443
D.5.3.3.1 Get Last Message Command 444
D.5.3.3.2 Message Confirmation Command 444
D.5.3.3.3 GetMessageCancellation Command 445
D.5.4 Application Guidelines 446
D.6 Tunneling Cluster 447
D.6.1 Overview 447
D.6.2 Server 450
D.6.2.1 Dependencies 450
D.6.2.2 Attributes 450
D.6.2.2.1 CloseTunnelTimeout Attribute 451
D.6.2.3 Parameters 451
D.6.2.4 Commands Received 451
D.6.2.4.1 RequestTunnel Command 452
D.6.2.4.2 CloseTunnel Command 453
D.6.2.4.3 TransferData Command 454
D.6.2.4.4 TransferDataError Command 455
D.6.2.4.5 AckTransferData Command 456
D.6.2.4.6 ReadyData Command 457
D.6.2.4.7 Get Supported Tunnel Protocols Command 458
D.6.2.5 Commands Generated 459
D.6.2.5.1 RequestTunnelResponse Command 459
D.6.2.5.2 TransferData Command 460
D.6.2.5.3 TransferDataError Command 461
D.6.2.5.4 AckTransferData Command 461
D.6.2.5.5 ReadyData Command 461
D.6.2.5.6 Supported Tunnel Protocols Response Command 461
D.6.2.5.7 TunnelClosureNotification Command 463
D.6.3 Client 463
D.6.3.1 Dependencies 463
D.6.3.2 Attributes 463
D.6.3.3 Commands Received 463
D.6.3.4 Commands Generated 463
D.7 Prepayment Cluster 465
D.7.1 Overview 465
D.7.2 Server 466
D.7.2.1 Dependencies 466
D.7.2.2 Attributes 466

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

512 D.7.2.2.1 Prepayment Information Attribute Set 467
513 D.7.2.2.2 Top-up Attribute Set 474
514 D.7.2.2.3 Debt Attribute Set 477
515 D.7.2.2.4 Supply Control Set 480
516 D.7.2.2.5 Alarms Attribute Set 480
517 D.7.2.2.6 Historical Cost Consumption Information Set 483
518 D.7.2.3 Commands Received 493
519 D.7.2.3.1 Select Available Emergency Credit Command 494
520 D.7.2.3.2 Change Supply Command 495
521 D.7.2.3.3 Change Debt Command 495
522 D.7.2.3.4 Emergency Credit Setup Command 497
523 D.7.2.3.5 Consumer Top Up Command 498
524 D.7.2.3.6 Credit Adjustment Command 499
525 D.7.2.3.7 Change Payment Mode Command 500
526 D.7.2.3.8 Get Prepay Snapshot Command 501
527 D.7.2.3.9 Get Top Up Log 502
528 D.7.2.3.10 Set Low Credit Warning Level 503
529 D.7.2.3.11 Get Debt Repayment Log Command 503
530 D.7.2.3.12 Set Maximum Credit Limit 504
531 D.7.2.3.13 Set Overall Debt Cap 505
532 D.7.2.4 Commands Generated 506
533 D.7.2.4.1 Supply Status Response Command 507
534 D.7.2.4.2 Publish Prepay Snapshot Command 507
535 D.7.2.4.3 Change Payment Mode Response Command 510
536 D.7.2.4.4 Consumer Top Up Response Command 511
537 D.7.2.4.5 Publish Top Up Log Command 513
538 D.7.2.4.6 Publish Debt Log Command 514
539 D.7.3 Client 515
540 D.7.3.1 Dependencies 515
541 D.7.3.2 Attributes 515
542 D.7.3.3 Commands Received 515
543 D.7.3.4 Commands Generated 515
544 D.7.4 Application Guidelines 515
545 D.7.4.1 Credit Status Attribute 515
546 D.7.4.1.1 Statuses Explained - an Example 517
547 D.8 Over-the-Air Bootload Cluster 520
548 D.8.1 Overview 520
549 D.8.2 OTA Bootloading Timing Considerations 520
550 D.9 Calendar Cluster 521
551 D.9.1 Overview 521
552 D.9.2 Server 524
553 D.9.2.1 Dependencies 524

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

554 D.9.2.2 Attributes 524
555 D.9.2.2.1 Auxiliary Switch Label Attribute Set 524
556 D.9.2.3 Commands Generated 525
557 D.9.2.3.1 PublishCalendar Command 525
558 D.9.2.3.2 PublishDayProfile Command 528
559 D.9.2.3.3 PublishWeekProfile Command 531
560 D.9.2.3.4 PublishSeasons Command 532
561 D.9.2.3.5 PublishSpecialDays Command 534
562 D.9.2.3.6 CancelCalendar Command 536
563 D.9.2.4 Commands Received 537
564 D.9.2.4.1 GetCalendar Command 537
565 D.9.2.4.2 GetDayProfiles Command 538
566 D.9.2.4.3 GetWeekProfiles Command 539
567 D.9.2.4.4 GetSeasons Command 540
568 D.9.2.4.5 GetSpecialDays Command 540
569 D.9.2.4.6 GetCalendarCancellation Command 541
570 D.9.3 Client 542
571 D.9.3.1 Dependencies 542
572 D.9.3.2 Attributes 542
573 D.9.3.3 Commands Received 542
574 D.9.3.4 Commands Generated 542
575 D.9.4 Application Guidelines 542
576 D.10 Device Management Cluster 544
577 D.10.1 Overview 544
578 D.10.1.1 Supplier Control 544
579 D.10.1.2 Tenancy Control 544
580 D.10.1.3 Password Control 544
581 D.10.2 Server 545
582 D.10.2.1 Dependencies 545
583 D.10.2.2 Attributes 545
584 D.10.2.2.1 Supplier Control Attribute Set 545
585 D.10.2.2.2 Tenancy Control Attribute Set 549
586 D.10.2.2.3 Backhaul Control Attribute Set 550
587 D.10.2.2.4 HAN Control Attribute Set 550
588 D.10.2.3 Commands Received 551
589 D.10.2.3.1 Get Change of Tenancy Command 552
590 D.10.2.3.2 Get Change of Supplier Command 552
591 D.10.2.3.3 RequestNewPassword Command 552
592 D.10.2.3.4 GetSiteID Command 553
593 D.10.2.3.5 Report Event Configuration Command 553
594 D.10.2.3.6 GetCIN Command 554
595 D.10.2.4 Commands Generated 554
596 D.10.2.4.1 Publish Change of Tenancy Command 555

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

D.10.2.4.2 Publish Change of Supplier Command 556
D.10.2.4.3 Request New Password Response Command 558
D.10.2.4.4 Update SiteID Command 559
D.10.2.4.5 SetEventConfiguration Command 560
D.10.2.4.6 GetEventConfiguration Command 562
D.10.2.4.7 Update CIN Command 562
D.10.3 Client 564
D.10.3.1 Dependencies 564
D.10.3.2 Attributes 564
D.10.3.2.1 Supplier Attribute Set 564
D.10.3.2.2 Price Event Configuration Attribute Set 565
D.10.3.2.3 Metering Event Configuration Attribute Set 569
D.10.3.2.4 Messaging Event Configuration Attribute Set 578
D.10.3.2.5 Prepayment Event Configuration Attribute Set 579
D.10.3.2.6 Calendar Event Configuration Attribute Set 583
D.10.3.2.7 Device Management Event Configuration Attribute Set 585
D.10.3.2.8 Tunnel Event Configuration Attribute Set 587
D.10.3.2.9 OTA Event Configuration Attribute Set 589
D.10.3.3 Commands Received 591
D.10.3.4 Commands Generated 591
D.10.4 Application Guidelines 591
D.10.4.1 Passwords 591
D.10.4.2 Consumer Password Use Case 591
D.10.4.3 Engineer Password Use Case 591
D.10.4.4 Password Security Recommendations 591
D.11 Events Cluster 592
D.11.1 Overview 592
D.11.2 Server 593
D.11.2.1 Dependencies 593
D.11.2.2 Attributes 594
D.11.2.3 Commands Received 594
D.11.2.3.1 Get Event Log Command 594
D.11.2.3.2 Clear Event Log Request Command 596
D.11.2.4 Commands Generated 597
D.11.2.4.1 Publish Event Command 597
D.11.2.4.2 Publish Event Log Command 598
D.11.2.4.3 Clear Event Log Response Command 600
D.11.3 Client 601
D.11.3.1 Dependencies 601
D.11.3.2 Attributes 601
D.11.3.3 Commands Received 601
D.11.3.4 Commands Generated 601

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

639 D.12 Energy Management Cluster 602
640 D.12.1 Overview 602
641 D.12.2 Server 602
642 D.12.2.1 Dependencies 602
643 D.12.2.2 Attributes 602
644 D.12.2.2.1 Load Control State Attribute 603
645 D.12.2.2.2 CurrentEventID Attribute 604
646 D.12.2.2.3 CurrentEventStatus Attribute 604
647 D.12.2.2.4 Conformance Level Attribute 606
648 D.12.2.2.5 MinimumOffTime, MinimumONTime & MinimumCyclePeriod Attributes 606
649 D.12.2.3 Attribute Reporting 607
650 D.12.2.4 Commands Received 607
651 D.12.2.4.1 Manage Event Command 607
652 D.12.2.5 Commands Generated 609
653 D.12.2.5.1 Report Event Status Command 609
654 D.12.3 Client 610
655 D.12.3.1 Dependencies 610
656 D.12.3.2 Attributes 610
657 D.12.3.3 Commands Received 610
658 D.12.3.4 Commands Generated 610
659 D.13 MDU Pairing Cluster 611
660 D.13.1 Overview 611
661 D.13.2 Server 611
662 D.13.2.1 Dependencies 611
663 D.13.2.2 Attributes 611
664 D.13.2.3 Commands Generated 611
665 D.13.2.3.1 Pairing Response Command 612
666 D.13.3 Client 613
667 D.13.3.1 Dependencies 613
668 D.13.3.2 Attributes 613
669 D.13.3.3 Commands Generated 613
670 D.13.3.3.1 Pairing Request Command 613
671 Annex E Rules And Guidelines For Overlapping Events 615
672 E.1 Definitions 615
673 E.2 Rules and Guideline 616
674 E.3 Event Examples 618
675 E.3.1 Correct Overlapping Events for Different Device Classes 619
676 E.3.2 Correct Superseded Event for a Device Class 620
677 E.3.3 Superseding Events for Subsets of Device Classes 621
678 E.3.4 Ending Randomization Between Events 622
679 E.3.5 Start Randomization Between Events 623
680 E.3.6 Acceptable Gaps Caused by Start and Stop Randomization of Events 624
681 Annex F Joining Procedure Using Pre-Configured Trust Center Link Keys 625

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

682 # List of Tables

683 Table 1-1 – Document Revision Change History 31
684 Table 5-1 – Startup Parameters 55
685 Table 5-2 – Join Parameters 57
686 Table 5-3 – Security Parameters 58
687 Table 5-4 – End Device Parameters 59
688 Table 5-5 – Link Status Parameters 59
689 Table 5-6 – Concentrator Parameters 59
690 Table 5-7 – APS Transport Parameters 60
691 Table 5-8 – APS Fragmentation Parameters 60
692 Table 5-9 – Binding Parameters 61
693 Table 5-10 – Per SE Network Storage Requirements 67
694 Table 5-11 – Example Hash of the TC Link Key 69
695 Table 5-12 – Parameters of Trust Center Swap-Out 70
696 Table 5-13 – Security Key Assignments per Cluster 73
697 Table 5-14 – Devices Specified in the Smart Energy Profile 95
698 Table 5-15 – Clusters Used in the Smart Energy Profile 97
699 Table 6-1 – Clusters Common to All Devices 102
700 Table 6-2 – Common Features and Functions Configuration for a Smart Energy Device 103
701 Table 6-3 – Clusters Supported by the Energy Service Interface 105
702 Table 6-4 – Clusters Supported by the Metering Device 106
703 Table 6-5 – Clusters Supported by the In-Home Display Device 108
704 Table 6-6 – Clusters Supported by the PCT 109
705 Table 6-7 – Clusters Supported by the Load Control Device 110
706 Table 6-8 – Clusters Supported by the Smart Appliance Device 111
707 Table 6-9 – Clusters Supported by the Prepayment Terminal Device 112
708 Table 6-10 – Clusters Supported by the Remote Communications Device 113
709 Table A-1– Additional Status Enumeration 115
710 Table A-2– *Attribute Reporting Status* Attribute 115
711 Table A-3– Attribute Reporting Status Enumerations 116
712 Table B-1– Parameters of the INTRP-DATA. request 120
713 Table B-2– Parameters of the INTRP-DATA. confirm 121
714 Table B-3– Parameters of the INTRP-DATA.indication 123
715 Table C-1– Clusters Specified for the Secure Communication Functional Domain 136
716 Table C-2– Key Establishment Attribute Sets 138
717 Table C-3– Key Establishment Attribute Sets 139
718 Table C-4– Values of the *KeyEstablishmentSuite* Attribute 139
719 Table C-5– Received Command IDs for the Key Establishment Cluster Server 139
720 Table C-6– *Terminate Key Establishment* Command Status Field 143
721 Table C-7– Key Establishment Attribute Sets 145

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Table C-8– Attributes of the Information Attribute Set 145
Table C-9– Values of the *KeyEstablishmentSuite* Attribute 146
Table C-10– Received Command IDs for the Key Establishment Cluster Client 146
Table C-11– *Terminate Key Establishment* Command Status Field 150
Table C-12– Values of the *KeyUsage* Field 157
Table C-13– ECC Implicit Certificate format 157
Table C-14– Parameters Used by Methods of the CBKE Protocol 159
Table D-1– Command IDs for the Demand Response and Load Control Server 190
Table D-2– Device Class Field BitMap/Encoding 191
Table D-3– Criticality Levels 192
Table D-4– Event Control Field BitMap 195
Table D-5– Cancel Control 197
Table D-6– Cancel All Command Cancel Control Field 198
Table D-7– Demand Response Client Cluster Attributes 199
Table D-8– Generated Command IDs for the Demand Response and Load Control Client 201
Table D-9– Event Status Field Values 202
Table D-10– Metering Cluster Server Attribute Sets 214
Table D-11– Reading Information Attribute Set 215
Table D-12– Block Enumerations 220
Table D-13– Supply Status Attribute Enumerations 221
Table D-14– TOU Information Attribute Set 224
Table D-15– Meter Status Attribute Set 228
Table D-16– Mapping of the *Status* Attribute (Electricity) 229
Table D-17– Meter *Status* Attribute (Gas) 229
Table D-18– Meter *Status* Attribute (Water) 230
Table D-19– Meter *Status* Attribute (Heat and Cooling) 230
Table D-20– General Flags of the Extended Status BitMap 232
Table D-21– Electricity -Meter specific Flags of the Extended Status BitMap 233
Table D-22– Gas-Meter specific Flags of the Extended Status BitMap 234
Table D-23– LowMediumHighStatus Attribute 235
Table D-24– Formatting Examples 236
Table D-25– Formatting Attribute Set 236
Table D-26– *UnitofMeasure* Attribute Enumerations 238
Table D-27– *MeteringDeviceType* Attribute 242
Table D-28– *TemperatureUnitOfMeasure* Enumeration 244
Table D-29– Historical Attribute Set 248
Table D-30– Load Profile Configuration Attribute Set 258
Table D-31– Supply Limit Attribute Set 259
Table D-32– Block Information Attribute Set (Delivered) 262
Table D-33– Alarm Attribute Set 266
Table D-34– Alarm Code Groups 267
Table D-35– Generic Alarm Group 268

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

764 Table D-36– Electricity Alarm Group 268
765 Table D-37– Generic Flow/Pressure Alarm Group 269
766 Table D-38– Water Specific Alarm Group 270
767 Table D-39– Heat and Cooling Specific Alarm Group 270
768 Table D-40– Gas Specific Alarm Group 270
769 Table D-41– Extended Generic Alarm Group 270
770 Table D-42– Manufacturer Specific Alarm Group 272
771 Table D-43– Block Information Attribute Set (Received) 272
772 Table D-44– Meter Billing Attribute Set 276
773 Table D-45– Supply Control Attribute Set 278
774 Table D-46– Alternative Historical Consumption Attribute Set 280
775 Table D-47– Generated Command IDs for the Metering Server 288
776 Table D-48– Status Field Values 290
777 Table D-49– ProfileIntervalPeriod Timeframes 290
778 Table D-50– Snapshot Schedule Confirmation 293
779 Table D-51– Snapshot Confirmation 294
780 Table D-52– Snapshot Cause BitMap 295
781 Table D-53– Snapshot Payload Type 296
782 Table D-54– Sample Type Enumerations 306
783 Table D-55– Notification Flags Order 311
784 Table D-56– Supply Status Field Enumerations 315
785 Table D-57 – Metering Cluster Client Attribute Sets 317
786 Table D-58 – Notification Attribute Set 317
787 Table D-59 – Functional Notification Flags 318
788 Table D-60 – Example Usage of Stay Awake Request Flags 319
789 Table D-61 – Push Historical Metering Data Definition 320
790 Table D-62 – Push Historical Payment Data Attribute Definition 320
791 Table D-63– Generated Command IDs for the Metering Client 322
792 Table D-64– Interval Channel Values 323
793 Table D-65– Snapshot Schedule BitMap 327
794 Table D-66– Supply Control Bits 333
795 Table D-67– Local Change Supply: Supply Status Field Enumerations 335
796 Table D-68– SetSupplyStatus: Field Enumerations 336
797 Table D-69– Notification Flags 2 345
798 Table D-70– Notification Flags 3 346
799 Table D-71– Notification Flags 4 347
800 Table D-72– Notification Flags 5 347
801 Table D-73– Price Cluster Attribute Sets 351
802 Table D-74– Tier Label Attribute Set 351
803 Table D-75– Block Threshold Attribute Set 353
804 Table D-76– Block Period Attribute Set 356

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Table D-77– *Commodity* Attribute Set 357
Table D-78– Values and Descriptions for the *CalorificValueUnit* Attribute 359
Table D-79– Block Price Information Attribute Set 360
Table D-80– Extended Price Information Set (TOU charging only) 364
Table D-81– Tariff Information Attribute Set 365
Table D-82– TierBlockMode Enumeration 366
Table D-83– CO_2Unit Enumeration 368
Table D-84– Billing Information Attribute Set 369
Table D-85– Credit Payment Attribute Set 370
Table D-86– CreditPaymentStatus Enumeration 371
Table D-87– PaymentDiscountDuration Enumerations 372
Table D-88– Received Tier Label Attribute Set 372
Table D-89– Received Block Threshold Attribute Set 373
Table D-90– Received Block Period Attribute Set 374
Table D-91– Received Block Price Attribute Set 374
Table D-92– Received Extended Price Information Attribute Set (TOU charging only) 375
Table D-93– Received Tariff Information Attribute Set 376
Table D-94– Received Billing Information Attribute Set 377
Table D-95– Received Command IDs for the Price Cluster 378
Table D-96– Generated Command IDs for the Price Cluster 393
Table D-97– Price Tier Sub-field Enumerations 396
Table D-98– Register Tier Sub-field Enumerations 398
Table D-99– Alternate Cost Unit Enumerations 399
Table D-100– Price Control Field BitMap 400
Table D-101– Generation Tier Enumerations 400
Table D-102– Extended Price Tier Field Enumerations 403
Table D-103– Extended Register Tier Field Enumerations 404
Table D-104– Block Period Control Field BitMap 407
Table D-105– Block Period DurationTimebase Enumeration 407
Table D-106– Block Period Duration Control Enumeration 408
Table D-107– Tariff Resolution Period Enumeration 408
Table D-108– Tariff Type Enumeration 412
Table D-109– Tariff Charging Scheme Enumeration 412
Table D-110– PublishPriceMatrix Sub-Payload Control Bitmap 415
Table D-111– PublishBlockThresholds Sub-Payload Control Bitmap 417
Table D-112– CPP Auth Enumeration 425
Table D-113– CPP Auth Enumeration 425
Table D-114– Currency Change Control 429
Table D-115– Price Client Cluster Attributes 431
Table D-116– Generated Command IDs for the Messaging Server 439
Table D-117– Message Control Field Bit Map 439
Table D-118– Extended Message Control Field Bit Map 441

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Table D-119– Messaging Client Commands 443
Table D-120– Message Confirmation Control 445
Table D-121– Tunneling Server Attributes 450
Table D-122– Cluster Parameters Passed Through Commands 451
Table D-123– Cluster -specific Commands Received by the Server 452
Table D-124– ProtocolID Enumerations 452
Table D-125– TransferDataStatus Values 456
Table D-126– Cluster-Specific Commands Sent by the Server 459
Table D-127– TunnelStatus Values 460
Table D-128– Prepayment Attribute Sets 466
Table D-129– P*repayment Information* Attribute Set 467
Table D-130– *Payment Control Configuration* Attribute 468
Table D-131– *Credit Status* Attribute 471
Table D-132– OverallDebtCap Example 471
Table D-133– Top-up Attribute Set 474
Table D-134– Debt Attribute Set 477
Table D-135– Debt Recovery Method Enumerations 479
Table D-136–*Recovery Frequency* Field Enumerations 479
Table D-137– Alarm Attribute Set 480
Table D-138– Prepayment Alarm Status Indicators 481
Table D-139– Alarms Code Group 482
Table D-140– PrepayGenericAlarmGroup 482
Table D-141– PrepaySwitchAlarmGroup 482
Table D-142– PrepayEventAlarmGroup 483
Table D-143– Historical Cost Consumption Information Attribute Set 483
Table D-144–*CurrencyScalingFactor* Enumerations 490
Table D-145– Cluster -specific Commands Received by the Server 493
Table D-146– Originating Device Field Enumerations 494
Table D-147– Debt Amount Type Field Enumerations 496
Table D-148– Credit Type Field Enumerations 499
Table D-149– Debt Type Field Enumerations 504
Table D-150– Cluster -specific Commands Sent by the Server 506
Table D-151– Snapshot Payload Cause 508
Table D-152– Snapshot Payload Type 509
Table D-153– Friendly Credit BitMap 510
Table D-154– Result Type Field Enumerations 512
Table D-155–Calendar Data Structures 522
Table D-156–Calendar Cluster Attribute Sets 524
Table D-157– Auxiliary Switch Label Attribute Set 524
Table D-158– Commands Generated by the Calendar Cluster Server 525
Table D-159– Calendar Type Enumeration 527

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Table D-160– Calendar Time Reference Enumeration 527
Table D-161– Commands Received by the Calendar Cluster Server 537
Table D-162– Device Management Attribute Sets 545
Table D-163– Supplier Control Attribute Set 545
Table D-164– Proposed Change Control 547
Table D-165– Contactor State Bit Combinations 548
Table D-166– Tenancy Control Attribute Set 549
Table D-167– Backhaul Control Attribute Set 550
Table D-168– State of the WAN Connection 550
Table D-169– HAN Control Attribute Set 550
Table D-170– Cluster Specific Commands Received by the Server 551
Table D-171– Cluster Specific Commands Generated by the Server 554
Table D-172– Password Type Enumeration 558
Table D-173– Configuration Control Enumeration 560
Table D-174– Device Management Client Attribute Sets 564
Table D-175– Supplier Attribute Sets 564
Table D-176– Price Event Configuration Attribute Set 565
Table D-177– Event Configuration Bitmaps 569
Table D-178– Metering Event Configuration Attribute Set 569
Table D-179– Messaging Event Configuration Attribute Set 578
Table D-180– Prepayment Event Configuration Attribute Set 579
Table D-181– Calendar Event Configuration Attribute Set 584
Table D-182– Device Management Event Configuration Attribute Set 585
Table D-183– Tunneling Event Configuration Attribute Set 587
Table D-184– OTA Event Configuration Attribute Set 589
Table D-185– Cluster Specific Commands Received by the Server 594
Table D-186– Log ID Enumeration 595
Table D-187– Event Control Bitmap 595
Table D-188– Cluster Specific Commands Generated by the Server 597
Table D-189– Event Action Control Bitmap 598
Table D-190– Log Payload Control Bitmap 599
Table D-191– ClearedEventsLogs Bitmap 600
Table D-192– Energy Management Server Cluster Attributes 602
Table D-193– Load Control State BitMap/Encoding 603
Table D-194– Current Event Status BitMap/Encoding 605
Table D-195– Received Command IDs for the Energy Management Server 607
Table D-196– Action(s) Required BitMap/Encoding 608
Table D-197– Generated Command IDs for the Energy Management Server 609
Table D-198– MDU Pairing Server Commands 611
Table D-199– MDU Pairing Client Commands 613

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

List of Figures

Figure 5-1 Utility Private HAN 51
Figure 5-2 Utility Private NAN 52
Figure 5-3 Customer Private NAN 52
Figure 5-4 An Example of a MDU Setup 53
Figure 5-5 Successful Join and Registration 72
Figure 5-6 Node Communication with Other Nodes on the Network Using APS Layer Encryption 77
Figure 5-7 Smart Energy Device Installation Code Process 78
Figure 5-8 Installation Code Use with the Trust Center 79
Figure B-1– ZigBee Stack with Stub APS 118
Figure B-2– Normal ZigBee Frame 124
Figure B-3– Enhanced Inter-PAN ZigBee Frame 125
Figure B-4– Stub NWK Header Format 125
Figure B-5– NWK Frame Control Field 125
Figure B-6– Stub APS Header Format 125
Figure B-7– APS Frame Control Field 126
Figure C-1– Overview of General Exchange 134
Figure C-2– Typical Usage of the Key Establishment Cluster 136
Figure C-3– Key Establishment Command Exchange 137
Figure C-4– *Initiate Key Establishment Request* Command Payload 140
Figure C-5– *Ephemeral Data Request* Command Payload 142
Figure C-6– *Confirm Key Request* Command Payload 142
Figure C-7– *Terminate Key Establishment* Command Payload 143
Figure C-8– *Initiate Key Establishment Response* 147
Figure C-9– *Ephemeral Data Response* Command Payload 148
Figure C-10– *Confirm Key Response* Command Payload 149
Figure C-11– *Terminate Key Establishment* Command Payload 150
Figure C-12– Key Establishment Command Exchange 165
Figure C-13– Key Establishment Command Exchange 177
Figure D-1– Demand Response/Load Control Cluster Client Server Example 189
Figure D-2– *Load Control Event* Command Payload 190
Figure D-3– *Cancel Load Control Event* Payload 196
Figure D-4– *Cancel All Load Control Events* Command Payload 198
Figure D-5– *Report Event Status* Command Payload 202
Figure D-6– *Get Scheduled Events* Command Payload 205
Figure D-7– Example of Both a Successful and an Overridden Load 209
Figure D-8– Example of a Load Curtailment Superseded and 210
Figure D-9– Standalone ESI Model with Mains Powered Metering Device 211
Figure D-10– Standalone ESI Model with Battery Powered Metering Device 212

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

969 Figure D-11– ESI Model with Integrated Metering Device 213
970 Figure D-12– *Get Profile Response* Command Payload 289
971 Figure D-13– *Request Fast Poll Mode Response* Command Payload 292
972 Figure D-14– *ScheduleSnapshotResponse* Command Payload 293
973 Figure D-15– Snapshot Response Payload Sub-Payload 293
974 Figure D-16– *TakeSnapshotResponse* Command Payload 294
975 Figure D-17– *Publish Snapshot* Command Payload 294
976 Figure D-18– Snapshot Utilizing Multiple Commands 297
977 Figure D-19– TOU Information Delivered Snapshot Sub-Payload 298
978 Figure D-20– TOU Information Received Snapshot Sub-Payload 299
979 Figure D-21– Block Information Delivered Snapshot Sub-Payload 300
980 Figure D-22– Block Information Received Snapshot Sub-Payload 301
981 Figure D-23– TOU Information Delivered (No Billing) Snapshot Sub-Payload 303
982 Figure D-24– TOU Information Received (No Billing) Snapshot Sub-Payload 303
983 Figure D-25– Block Information Delivered (No Billing) Snapshot Sub-Payload 304
984 Figure D-26– Block Information Received (No Billing) Snapshot Sub-Payload 305
985 Figure D-27– Format of the *GetSampledDataResponse* Command Payload 306
986 Figure D-28– Format of the *ConfigureMirror* Command Payload 307
987 Figure D-29– *MirrorReportAttributeResponse* Command Enabled 308
988 Figure D-30– *MirrorReportAttributeResponse* Command Disabled 309
989 Figure D-31– *NotificationScheme* Enumerations 310
990 Figure D-32– *ConfigureNotificationScheme* Command Payload 311
991 Figure D-33– *ConfigureNotificationFlags* Command Payload 312
992 Figure D-34– *Bit Field Allocation* Command Sub Payload 313
993 Figure D-35– *GetNotifiedMessage* Command Payload 314
994 Figure D-36– Supply Status Response Command Payload 315
995 Figure D-37– *StartSamplingResponse* Command Payload 316
996 Figure D-38– *Get Profile* Command Payload 323
997 Figure D-39– *Request Mirror Response* Command Payload 324
998 Figure D-40– *Mirror Removed* Command Payload 325
999 Figure D-41– *Request Fast Poll Mode* Command Payload 325
1000 Figure D-42– *ScheduleSnapshot* Command Payload 326
1001 Figure D-43–*SnapshotSchedulePayload* Format 326
1002 Figure D-44– *TakeSnapshot* Command Payload 327
1003 Figure D-45– *GetSnapshot* Command Payload 328
1004 Figure D-46– *StartSampling* Command Payload 329
1005 Figure D-47– *GetSampledData* Command Payload 330
1006 Figure D-48– MirrorReportAttributeResponse Command Payload 331
1007 Figure D-49– *ResetLoadLimitCounter* Command Payload 332
1008 Figure D-50– Change Supply Command Payload 333
1009 Figure D-51– Local Change Supply Command Payload 335
1010 Figure D-52– SetSupplyStatus Command Payload 336
1011 Figure D-53– SetUncontrolledFlowThreshold Command Payload 337

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

1012 Figure D-54– Example of Data flow from IHD to Gas meter 342
1013 Figure D-55– Price Cluster Client Server Example 350
1014 Figure D-56– Single Threshold Set applied to All Consumption 367
1015 Figure D-57– Single Threshold Set applied to Each Tier Consumption 367
1016 Figure D-58– *GetCurrentPrice* Command Payload 379
1017 Figure D-59– *Get Current Price* Command Options Field 380
1018 Figure D-60– *GetScheduledPrices* Command Payload 381
1019 Figure D-61– *PriceAcknowledgement* Command Payload 381
1020 Figure D-62– *Get Block Period(s)* Command Payload 382
1021 Figure D-63– *GetConversionFactor* Command Payload 383
1022 Figure D-64– *GetCalorificValue* Command Payload 384
1023 Figure D-65– GetTariffInformation Command Payload 385
1024 Figure D-66– GetPriceMatrix Command Payload 386
1025 Figure D-67– GetBlockThresholds Command Payload 387
1026 Figure D-68– GetCO_2Value Command Payload 387
1027 Figure D-69– GetTierLabels Command Payload 388
1028 Figure D-70– GetBillingPeriod Command Payload 389
1029 Figure D-71– GetConsolidatedBill Command Payload 390
1030 Figure D-72– CPPEventResponse Command Payload 391
1031 Figure D-73– *GetCreditPayment* Command Payload 392
1032 Figure D-74– *Publish Price* Command Payload 395
1033 Figure D-75– *Publish Block Period* Command Payload 406
1034 Figure D-76– *PublishConversionFactor* Command Payload 409
1035 Figure D-77– *PublishCalorificValue* Command Payload 410
1036 Figure D-78– *PublishTariffInformation* Command Payload 411
1037 Figure D-79– *PublishPriceMatrix* Command Payload 414
1038 Figure D-80– *PriceMatrix* Command Sub-Payload 415
1039 Figure D-81– *PublishBlockThresholds* Command Payload 416
1040 Figure D-82– BlockThreshold Sub-Payload Format 417
1041 Figure D-83– *PublishCO_2Value* Command Payload 419
1042 Figure D-84– *PublishTierLabels* Command Payload 420
1043 Figure D-85– *PublishBillingPeriod* Command Payload 421
1044 Figure D-86– *PublishConsolidatedBill* Command Payload 423
1045 Figure D-87– *PublishCPPEvent* Command Payload 424
1046 Figure D-88– CPP Event Flow 426
1047 Figure D-89– *PublishCreditPayment* Command Payload 427
1048 Figure D-90– *PublishCurrencyConversion* Command Payload 428
1049 Figure D-91– *CancelTariff* Command Payload 430
1050 Figure D-92– Messaging Cluster Client/Server Example 438
1051 Figure D-93– *Display Message* Command Payload 439
1052 Figure D-94– *Cancel Message* Command Payload 442
1053 Figure D-95– *Cancel All Messages* Command Payload 443

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Figure D-96– *Message Confirmation* Command Payload 444
Figure D-97– Get*MessageCancellation* Command Payload 445
Figure D-98– Client/Server Message Command Exchanges 446
Figure D-99– A Client Requests a Tunnel From a Server to Exchange Complex 447
Figure D-100– SE Device 1 (Client) Requests a Tunnel From SE Device 2 (Server) to Transfer Data Without Flow Control (Default) 449
Figure D-101– SE Device 1 (Client) Requests a Tunnel From SE Device 2 (Server) to Transfer Data With Flow Control 450
Figure D-102– *RequestTunnel* Command Payload 452
Figure D-103– *CloseTunnel* Command Payload 454
Figure D-104– *TransferData* Command Payload 454
Figure D-105– *TransferDataError* Command Payload 455
Figure D-106– *AckTransferData* Command Payload 456
Figure D-107– *ReadyData* Command Payload 457
Figure D-108– *Get Supported Tunnel Protocols* Command Payload 458
Figure D-109– *RequestTunnelResponse* Command Payload 459
Figure D-110– *TransferData* Command Payload 461
Figure D-111– *Supported Tunnel Protocols Response* Command Payload 462
Figure D-112– *Supported Tunnel Protocols Response* Command Protocol Fields 462
Figure D-113– *TunnelClosureNotification* Command Payload 463
Figure D-114– Prepay Cluster Client Server Example 466
Figure D-115– *Select Available Emergency Credit* 494
Figure D-116– Change Debt Command Payload 495
Figure D-117– Emergency Credit Setup Command Payload 497
Figure D-118– Consumer Top Up Command Payload 498
Figure D-119– Credit Adjustment Command Payload 499
Figure D-120– Change Payment Mode Command Payload 500
Figure D-121– Get Prepay Snapshot Command Payload 501
Figure D-122– Get Top Up Code Log Command Payload 502
Figure D-123– Set Low Credit Warning Level Command Payload 503
Figure D-124– GetDebtRepaymentLog Command Payload 503
Figure D-125– Set Maximum Credit Level Command Payload 505
Figure D-126– Set Overall Debt cap Command Payload 506
Figure D-127– Publish Prepay Snapshot Command Payload 507
Figure D-128– Debt/Credit Status Command Payload 509
Figure D-129– Change Payment Mode Response Command Payload 510
Figure D-130– Consumer Top Up Response Command Payload 511
Figure D-131– Publish Top Up Log Command Payload 513
Figure D-132– Top Up Code Payload 513
Figure D-133– Publish Debt Log Command Payload 514
Figure D-134– Format of a Debt Payload Record 514
Figure D-135– Prepayment Credit Status Attribute Explained 517
Figure D-136–Calendar Cluster 521

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

1097 Figure D-137– Recommended Calendar Command Sequence 523
1098 Figure D-138– *PublishCalendar* Command Payload 526
1099 Figure D-139– *PublishDayProfile* Command Payload 529
1100 Figure D-140– Schedule Entries for Rate Start Times Command Sub-Payload 530
1101 Figure D-141– Schedule Entries for Friendly Credit Start Times Command Sub-Payload 530
1102 Figure D-142– Schedule Entries for Auxilliary Load Start Times Command Sub-Payload 531
1103 Figure D-143– *PublishWeekProfile* Command Payload 532
1104 Figure D-144– *PublishSeasons* Command Payload 533
1105 Figure D-145– Season Entry Sub-Payload 534
1106 Figure D-146– *PublishSpecialDays* Command Payload 535
1107 Figure D-147– *SpecialDayEntry* Sub-Payload 536
1108 Figure D-148– *CancelCalendar* Command Payload 536
1109 Figure D-149– GetCalendar Command Payload 538
1110 Figure D-150– GetDayProfiles Command Payload 539
1111 Figure D-151– GetWeekProfiles Command Payload 539
1112 Figure D-152– GetSeasons Command Payload 540
1113 Figure D-153– GetSpecialDays Command Payload 541
1114 Figure D-154– Device Management Cluster Client/Server Example 544
1115 Figure D-155– RequestNewPassword Command Payload 552
1116 Figure D-156– Report Event Configuration Command Payload 553
1117 Figure D-157– Report Event Configuration Sub-Payload 554
1118 Figure D-158– Publish Change of Tenancy Command Payload 555
1119 Figure D-159– Publish Change of Supplier Command Payload 556
1120 Figure D-160– RequestNewPasswordResponse Command Payload 558
1121 Figure D-161– Update SiteID Command Payload 559
1122 Figure D-162– Set Event Configuration Command Payload 560
1123 Figure D-163– ‘Apply by List’ Sub-Payload 561
1124 Figure D-164– ‘Apply by Event Group’ Sub-Payload 561
1125 Figure D-165– ‘Apply by Log Type’ Sub-Payload 561
1126 Figure D-166– ‘Apply by Configuration Match’ Sub-Payload 562
1127 Figure D-167– Get Event Configuration Command 562
1128 Figure D-168– Update CIN Command Payload 563
1129 Figure D-169– Event Cluster Client Server Example 592
1130 Figure D-170– Mirrored BOMD Event Cluster Client Server Example 593
1131 Figure D-171– Get Event Log Command Payload 595
1132 Figure D-172– Clear Event Log Request Command Payload 596
1133 Figure D-173– Publish Event Command Payload 597
1134 Figure D-174– Publish Event Log Command Payload 598
1135 Figure D-175– Publish Event Log Sub-Payload 599
1136 Figure D-176– Clear Event Log Response Command Payload 600
1137 Figure D-177– Energy Management Cluster 602
1138 Figure D-178– *Manage Event* Command Payload 607

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Figure D-179– *Report Event Status* Command Payload 609
Figure D-180– MDU Pairing Cluster 611
Figure D-181– Pairing Response Command Payload 612
Figure D-182– Pairing Request Command Payload 614
Figure E-1– Smart Energy Device Class Reference Example 618
Figure E-2– Correctly Overlapping Events 619
Figure E-3– Correct Superseding of Events 620
Figure E-4– Superseded Event for a Subset of Device Classes 621
Figure E-5– Ending Randomization Between Events 622
Figure E-6– Start Randomization Between Events 623
Figure E-7– Acceptable Gaps with Start and Stop Randomization 624

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Participants

The following is a list of those who were members of the ZigBee PRO Technical Sub-committee (PRO-TSC) leadership when this document was released:

Ezra Hale: Chair

Phil Jamieson: Vice-Chair

When the document was released, the ZigBee PRO Energy Domain Working Group leadership was composed of the following members:

Larry Kohrmann: Chair

Ian Winterburn: Vice-Chair

Rob Alexander & Ian Winterburn: Technical Editors

Contributions were made to this document from the following members:

Rob Alexander	Tim Enwall	Rolf Kistler	Jeff Shudark
Skip Ashton	Tony Field	John Knuth	Robby Simpson
Arvind Asthana	Chris Fitzer	Zin Kyaw	Sumit Singh
Wally Barnum	Chris Gales	Chris LaPre	David Smith
Vincent Bemmel	Benjamin Gartner	Christopher Leidigh	Matt Smith
Gary Birk	Daniel Gavelle	Bill Lichtensteiger	Zachary Smith
Rick Bojahra	Dean van Gerrevink	John Lin	Mike Strumik
Mark Borins	Tim Gillman	Dan Lohman	Michael G. Stuber
John Buffington	Sean Gulliford	Juan Agüí Martín	Don Sturek
Jeff Cooper	Jesper Hæe	Tom Martin	Jakob Thomsen
Damon Corbin	Matt Hamblin	Jeff Mathews	Jesper Thomsen
Michael Cowan	Elias Hanna	Tony Mauro	MichelVeillette
John Cowburn	Jim Hartman	Luca Negri	MadsWestergreen
Robert Cragie	Donald Hasson	Howard Ng	Richard White
Jonathan Cressman	Heinz Hohl	Eugene Peng	IanWinterburn
Jeff Drake	Jeff Hugghins	John Prater	Jack Worsnop
Rick Enns	Jeffrey King	Yuri Shteinman	

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Document History

Table 1.1 shows the change history for this specification.

Table 1-1 – Document Revision Change History

Revision	Version	Description
0		Original version.
1		First draft to include annexes and cluster information.
2		Updated to include Key Establishment Cluster Annex. Added other minor changes within the document.
3		Included comments from internal Smart Energy (formerly Smart Energy) group review. New Items: Added Power Factor in Simple Metering cluster. Added Group support in the DR/LC cluster.
4		Included comments from internal Smart Energy group review scattered throughout. A number of field and attribute adds to the clusters.
5		Corrected Document Number issues, otherwise same as Revision 4.
6		Additional changes: Usual grammar and spelling changes *Load Profile* commands have been updated. Added Attributes to support the latest partial LP interval. Load Control rules for DR/LC Randomization ESP Historical Attributes. changes in Simple Metering cluster Changes to the *Get Current Price* command
7		Grammar, spelling, and formatting changes
8		PDF version of 07
9		First pass at comment resolution. Please refer to document #075424r03ZB for changes.
10		Second pass at comment resolution. Please refer to document #075424r04ZB for changes. Renamed to Smart Energy Profile.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

11		Third pass at comment resolution. Please refer to document #075424r05ZB for changes. Moved SE Cluster definitions into the annex D. Significant changes in the Security related sections. Best practice information added to more sections. Updated Annex E covering overlapping event examples. Corrected issues relating to the following CCBs: CC-900 [SE] Randomizing Price Events CC-901 [SE] Message Cluster Start Time CC-902 [SE] New Status Field for *Get Profile Response* Command CC-903 [SE] Support for Binding CC-904 [SE] ESP Historical Consumption Attributes in Simple Metering Server CC-905 [SE] More Precise Event Status Enumeration for *Report Event Status* Command CC-906 [SE] Additional Description of Device Class bits 0 and 1 CC-907 [SE] Array vs. Series of Intervals for *Get Profile Response* Command CC-908 [SE] Randomization of *Report Event Status* Command Send Times CC-909 [SE] Effective Time Field of Cancel Load Control Event to be Mandatory CC-910 [SE] Consolidation of Joining Procedures CC-911 [SE] Out of Bands Methods of Authentication CC-912 [SE] Method to Make Registered Devices Listed on ESP CC-913 [SE] Clarification of Rate Label Field of *Publish Price* Command.
12		PDF version of 075356r11.
13		Converted from Word to FrameMaker, includes all CCBs called out in the SE Profile Errata 08119r08.
14		Final editorial changes for initial publication.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

15		Corrected issues related to the following CCBs (from errata document 084914r05): CC-964 ZigBee Cluster Library reference doesn't contain the revision number. CC-965 Specification needs to clarify the service discovery process steps prior to and after the Key Establishment process. End Devices must also initiate the processes. CC-966 The Identify cluster should be Optional, not mandatory. CC-967 The Common Features and Functions table incorrectly calls out the binding and service discovery requests as mandatory items. CC-968 Future definitions of fields added to the end of commands are to be treated as reserved fields. CC-973 Addition of Greenhouse Gas (CO_2) pricing information to the *Publish Price* command. CC-974 Addition of Supply Limit tracking in the Metering Cluster. CC-980 Correct and describe CRC Algorithm used for Installation Codes. CC-981 Correct the Installation Codes text examples and provide example source code for testing/using the MMO Hash Algorithm. CC-982 Attributes *CurrentPartialProfileIntervalValueDelievered* and *CurrentPartialProfileIntervalValueReceived* do not list default values or mandatory/optional status. CC-983 Attributes *Power Factor*, *ReadingSnapShotTime*, *CurrentMaxDemandDelieveredTime*, and *CurrentMaxDemandReceivedTime* are incorrectly replicated in another section. Corrected issues related to the following CCBs: CC-984 Addition of Key Establishment test vectors. CC-986 Addition of metering device types to the simple metering cluster attribute *MeteringDeviceType Enumeration*. CC-993 Initiate Key Establishment Request and Response Payload Format field names need to match field names defined in Payload Format figures.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

16		CC-923 Best Practices for Client devices using the Inter-PAN Transmission section (Annex B) CC-940 Rename Simple Metering to just Metering CC-996 Test & Profile Specification Conflict (Message Confirmation) CC-1002 Typos (*Publish Price* Command Start Time description.) CC-1015-ESP Historical Consumption CC-1018 Mirror Device CC-1026 Remove term “unsecure rejoin” from document. CC-1027 Messaging cluster message payload size CC-1028 Add DRLC Commands Received to D.2.2 CC-1030 Range Extender does not allow support of optional clusters. CC-1031 Price Cluster client server references transposed CC-1032 Price Server Cluster Attributes CC-1059 Extra word CC-1060 Rijndael source code URL no longer valid CC-1069 Update reference to ZCL specification (075123r02) CC-1070 Publish Price payload format clarification and value of unused optional fields. CC-1072 ZigBee Smart Energy naming of ESI CC-1077 Interval Channel data type CC-1082 Range of InstantaneousDemand CC-1083 Price cluster clarifications CC-1087 Price clients cannot request all price values CC-1090 Example string is too long CC-1096 Message Confirmation Payload Details Typo CC-1098 Key establishment confirm key response CC-1103 Demand limit enabled or not CC-1108 Allow more flexibility with Issuer Event ID

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

16 cont'd		CC-1118 Recommended Practices for devices in a multi-ESP HAN CC-1119 Responding to a Get message when server's list is empty CC-1124 Add mcf Unit of Measure CC-1125 Clarify the text describing Issuer field and known CA CC-1130 Binding Clarification CC-1135 Change Attribute Access CC-1159 CMU Resolve CMU Audit response editorial comments CC-1160 Resolve CMU Audit response technical comments CC-1170 Add unitless Unit of Measure CC-1173 Duty Cycle Proposal CC-1179 Alarms Cluster support in Smart Energy Metering Device CC-1180 Addition of MJ (Mega Joule) Incremental Release 1 CCB 1181 Addition of optional attributes to the metering cluster required for district heat and cooling metering CCB 1195 Signature on DRLC messages are unused and should not be required CCB 1198 Data Type of MeteringDeviceType CCB 1206 When is the Price Acknowledegment Generated? CCB 1207 Is Price Acknowledgement optional? CCB 1210 Optional Alternate Cost Attribute do not specify unused defaults CCB 1244 Meaning of Start Time in Get Scheduled Events is ambiguous CCB 1320 Final SE 1.1 Interop Issues CCB 1389 Event ID backwards compatibility issue

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

17		CCB 994 - InterPAN Messaging should not allow any message CCB 1217 - Clarification on access control for Mirroring CCB 1218 - Prepayment cluster should also be mirrored CCB 1219 - Add push (report attribute) to align with Mirroring section later on CCB 1226 - Naming of SE Display Device CCB 1241 - Trust Center Swap-Out not explicitly optional CCB 1243 - Adjusting events with Start Time = 0x00000000 CCB 1258 - Need modifications to SE 1.1 specification for OTA CCB 1262 - Mirror CCB 1264 - Add support for CV and PTZ (gas conversion factors) to Price cluster CCB 1265 - Improvements to the Handling of Multiple Fuels CCB 1267 - Change End Device restriction from a SHALL to a SHOULD CCB 1268 - Price Tier Sub-fields are not sequential CCB 1269 - Multiplier and Divisor attributes should also apply to newly added EnergyCarrier and Temperature attributes

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

17 cont'd		CCB 1270 - Temperature attributes ought to be signed 24 bit integers CCB 1273 - Addition of 'Get Supported Tunnel Protocols' Command and Response CCB 1284 - Start-up Parameter TC Address allows non-Coordinator CCB 1289 - PhysicalEnvironment bit for Mirroring CCB CCB 1292 - OctetString payload octet counts incorrect CCB CCB 1293 - Start/Stop randomization for DRLC and Price CCB CCB 1294 - Miscellaneous editorial comments CCB 1300 - ThreshholdMultiplier behavior when 0 needs definition CCB 1322 - Inconsistent spelling of Enrolment CCB 1324 - SignatureType and Signature not marked Optional (O) in Figure D.5 CCB 1332 - Price Tier CCB 1334 - "Publish Price" typo as "Public Price" command CCB 1339 - Event Override CCB 1341 - kW and kWh in table D.22 CCB 1347 - Clarification on Publish Price command CCB 1349 - Time client should be allowed on ESI CCB 1350 - Add recommended practice for time synchronizing ESIs CCB 1352 - Modification of Multi-ESI mechanism from inter-op event CCB 1353 - Tunneling cluster transfer size establishment is incomplete CCB 1355 - Tunneling cluster CloseTimeout attribute should have special behavior when set to 0 CCB 1376 - Metering Device Types CCB 1380 - Unclear what happens when a received LCE event is "ignored" CCB 1382 - Typo in text CCB 1383 - Commodity Type to be read after service discovery to understand type of Price server

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

17 cont'd		CCB 1384 - Wrong data type for Supply Status CCB 1389 - Event ID backwards compatibility issue CCB 1397 - Clarify cluster usage of security keys CCB 1398 - Add a Signature Type of None CCB 1401 - Add TunnelClosureNotification command CCB 1403 - Permit Join Best Practices CCB 1404 - TC behavior unclear when devices leave network CCB 1419 - Trust Center Swap-out - Bindings & Mirrors CCB 1437 - DeviceClass is marked read-writeable but write may not be allowed CCB 1440 - Mirroring Feature CCB 1452 - Visibility of Mirror Endpoints CCB 1486 - End point requirement

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

18	1.1b	CCB 1275 - Extended PAN ID CCB 1276 - CCB 996 CCB 1283 - IPD required clusters draft text CCB 1285 - EUI64 mandatory in TC swap-out procedure CCB 1286 - Conflicting requirements for join behaviors CCB 1316 - DeviceClass Value Type/Range CCB 1318 - DRLC Server Does Not Specify Minimum Number of Events CCB 1325 - Number of Events incorrectly references GetBlockPeriod command CCB 1333 - Range of Price Ratio, Generation Price Ratio CCB 1346 - Heating/Cooling Set Point Ranges CCB 1348 - Response to Cancel Load Control Event CCB 1441 - Average Load Adjustment Percentage Text Appears to be Incorrect CCB 1449 - Actions Taken before Time Synced CCB 1455 - DRLC Cancel Load Control - Ignore Effective Time CCB 1456 - DRLC Cancel Load Control with different filters than the one creating the event. CCB 1457 - IHDs shall have the Device Types of the DRLC Events they wish to Display CCB 1482 - Conflicting Behavior for TC Swap-out and Rejoin when using Installation Codes CCB 1491 - Trust Center Keep Alive messages must be APS encrypted CCB 1494 - Add Billing Period Attribute Set to the Price Cluster [RIB]

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

18 cont’d		CCB 1500 - New Metering Attribute for Block Pricing [RIB] CCB 1537 - Previous Changes added Requirements to Incorrect Section [RIB] CCB 1547 - Consolidate Block Pricing Requirements [RIB] CCB 1564 - Unclear on the IPD cluster requirement CCB 1570 - Rename Display Device to "In-Home Display" (IHD) CCB 1572 - Missing Document History CCB 1592 - Missing word in sentence makes spec confusing Minor typographical corrections resulting from the document ballot

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

19	1.2a	Changes resulting from the SE1.2 MRD (11-5543) and TRD (11-5758), together with the 'Blackwall' changes identified in 14-0046. These include:- - Significant changes to Metering, Price and Messaging clusters - Prepayment cluster rewritten - Calendar, Device Management and Events clusters added - Federated Trust Center, including the MDU Pairing cluster added (**not currently certifiable**) - Energy Management cluster added (**not currently certifiable**) - Addition of the 128-bit 'sect283k1' ECC curve - Removal of the 'Anonymous Inter-PAN' mechanism and replacement with an 'Enhanced' Inter-PAN mechanism (**not currently certifiable**) - Remote Communications Device type added - Coexistence with Devices using other Profiles (**not currently certifiable**) CCB 1501 - All ZCL(general) clusters except those in Table 5.13 should be definitively prescribed to use a specific security key CCB 1511 - Tier Label Attribute Set CCB 1679 - Ambiguous definition for when block n vs block n+1 is active CCB 1746 - Incorrect Maximum TC Keep-Alive Interval CCB 1886 - IntervalReadReportingPeriod - Remove Attribute from the Spec CCB 1939 - TunnelID reference in RequestTunnel command should be ProtocolID CCB 1999 - Heating and Cooling Mandatory Attributes CCB 2009 - Publish Topup Command CCB 2010 - Handling of Incorrect Issuer Event ID Fields

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

1 Introduction

1.1 Scope

This Standard defines device descriptions and standard practices for "Smart Energy" applications needed in a Smart Energy based residential or light commercial environment. Installation scenarios range from a single home to an entire apartment complex. The key application domains are metering, pricing (with associated scheduling), messaging and demand response and load control. Support applications are provided for commissioning and managing such networks. Provision is made for operating in both Credit and Prepayment modes. Other applications will be added in future versions.

1.2 Purpose

This specification provides standard interfaces and device definitions to allow interoperability among ZigBee devices produced by various manufacturers of electrical equipment, meters, and Smart Energy enabling products.

1.3 Provisional Features

Some of the features in this version of this specification are provisional and non-certifiable. The text regarding these features may change before reaching certifiable status. The features consist of the following items:

- Metering cluster attribute 0x0012 (*VolumePerReport*)
- Configuration of Notification Schemes/Flags within the Metering cluster
- The Price cluster client attributes.
- Price cluster Critical Peak Pricing (CPP)
- Messaging cluster *Cancel All Messages* and *GetMessageCancellation* commands
- Tunneling cluster Flow Control option.
- Trust Center Swapout behaviors.
- Energy Management cluster.
- Federated Trust Center, including the MDU Pairing Cluster
- Coexistence with devices using other Profiles
- 'Enhanced' Inter-PAN transmission mechanism

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

2 References

2.1 References

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

2.1.1 ZigBee Alliance Documents

[B1] ZigBee document 07-5123-04, ZigBee Cluster Library Specification, ZigBee Cluster Library Development Board.

[B2] ZigBee document 064309r04, Commissioning Framework

[B3] ZigBee Document 05-3474-20, The ZigBee Specification, ZigBee Technical Steering Committee (TSC)

[B4] ZigBee Document 03084r00, ZigBee Key Establishment Proposal Certicom

[B5] ZigBee 075297r04, Proposal for Inter-PAN Exchange of Data in ZigBee

[B6] ZigBee document 095343r01, Installation Code Sample Source Code

[B7] ZigBee document 08006r05, ZigBee 2007 Layer PICS and Stack Profiles, ZigBee Core Stack Working Group

[B8] Over the Air Upgrade Cluster Spec. 09-5264-23

[B9] Over the Air Upgrade Cluster test spec: 09-5473-07

[B10] Over the Air Upgrade Cluster PICs: 09-5284-09

2.1.2 External Reference Documents

[B11] Institute of Electrical and Electronics Engineers, Inc., IEEE Std. 802.15.4 2003, IEEE Standard for Information Technology Telecommunications and Information Exchange between Systems - Local and Metropolitan Area Networks - Specific Requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (WPANs). New York: IEEE Press. 2003

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

[B12] ANSI X9.62-2005, Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA), American Bankers Association. Available from http://www.ansi.org.

[B13] ANSI X9.63-2001, Public Key Cryptography for the Financial Services Industry - Key Agreement and Key Transport Using Elliptic Curve Cryptography, American Bankers Association, November 20, 2001.
Available from http://www.ansi.org.

[B14] NIST Special Publication 800-56A, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography (Revised), March 2007.
Available from http://csrc.nist.gov.

[B15] NIST Special Publication 800-38C, Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality, May 2004.
Available from http://csrc.nist.gov.

[B16] FIPS Pub 197, Advanced Encryption Standard (AES), Federal Information Processing Standards Publication 197, US Department of Commerce/N.I.S.T, Springfield, Virginia, November 26, 2001.
Available from http://csrc.nist.gov.

[B17] FIPS Pub 198, The Keyed-Hash Message Authentication Code (HMAC), Federal Information Processing Standards Publication 198, US Department of Commerce/N.I.S.T., Springfield, Virginia, March 6, 2002.
Available from http://csrc.nist.gov.

[B18] Standards for Efficient Cryptography: SEC 1 (working draft) ver 1.7: Elliptic Curve Cryptography, Certicom Research, November 13, 2006.
Available from http://www.secg.org

[B19] Standards for Efficient Cryptography: SEC 4 (draft) ver 1.0: Elliptic Curve Cryptography, Certicom Research, January 24, 2013.
Available from http://www.secg.org

[B20] RFC 3280: Internet X.509 Public Key Infrastructure: Certificate and Certificate Revocation List (CRL) Profile. IETF, April 2002.
Available from http://www.ietf.org

[B21] DUPLICATE OF [B19] - REMOVED

[B22] RFC 3280: Internet X.509 Public Key Infrastructure: Certificate and Certificate Revocation List (CRL) Profile. IETF, April 2002. Available from http://www.ietf.org

[B23] Standards for Efficient Cryptography: SEC 2 (Final) ver 2.0: Recommended Elliptic Curve Domain Parameters

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

3 Definitions

3.1 Conformance Levels

Expected: A key word used to describe the behavior of the hardware or software in the design models assumed by this Standard. Other hardware and software design models may also be implemented.

May: A key word indicating a course of action permissible within the limits of the standard ("may" equals "is permitted").

Shall: A key word indicating mandatory requirements to be strictly followed in order to conform to the standard; deviations from shall are prohibited ("shall" equals "is required to").

Should: A key word indicating that, among several possibilities, one is recommended as particularly suitable, without mentioning or excluding others; that a certain course of action is preferred but not necessarily required; or, that (in the negative form) a certain course of action is deprecated but not prohibited ("should" equals "is recommended that").

3.2 ZigBee Definitions

Attribute: A data entity which represents a physical quantity or state. This data is communicated to other devices using commands.

Cluster: A container for one or more attributes and/or messages in a command structure.

Cluster identifier: A reference to the unique enumeration of clusters within a specific application profile. The cluster identifier is a 16-bit number unique within the scope of the application profile and identifies a specific cluster. Cluster identifiers are designated as inputs or outputs in the simple descriptor for use in creating a binding table.

Device: A description of a specific device within an application profile. For example, the light sensor device description is a member of the home automation application profile. The device description also has a unique identifier that is exchanged as part of the discovery process.

Node: Same as a unit.

Product: A product is a unit that is intended to be marketed. It implements application profiles that may be a combination of private, published, and standard.

Service discovery: The ability of a device to locate services of interest.

Unit: A unit consists of one or more physical objects (e.g., switch, controller, etc.) and their corresponding application profile(s) that share a single 802.15.4 radio. Each unit has a unique 64-bit IEEE address.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

ZigBee coordinator: An IEEE 802.15.4-2003 PAN coordinator.

ZigBee end device: an IEEE 802.15.4-2003 RFD or FFD participating in a ZigBee network, which is neither the ZigBee coordinator nor a ZigBee router.

ZigBee router: an IEEE 802.15.4-2003 FFD participating in a ZigBee network, which is not the ZigBee coordinator but may act as an IEEE 802.15.4-2003 coordinator within its personal operating space, that is capable of routing messages between devices and supporting associations.

3.3 Smart Energy Definitions

Delivered: In the context of metering, the term "Delivered" refers to the quantity of a commodity that was delivered to the customer from the utility.

Export: In the context of metering, the term "Export" has the same meaning as "Received".

Friendly Credit Period: A duration of time where delivery rules for supply interruption connected with the accounting functions of a meter are temporarily suspended. Also known as "Non-Disablement Period".

Note1 - When a meter enters into a Friendly Credit Period with a useable positive credit balance, the consumer will be allowed to consume energy for the duration of the Friendly Credit Period, regardless of their credit status while in that period. If, however, the consumer had already run out of credit and supply was interrupted before entering into the Friendly Credit Period, they will not be allowed to reconnect without first adding suitable additional credit.

Note2 - At the end of the Friendly Credit Period, the normal delivery rules connected with the accounting functions of the meter will be resumed, and if the meter's credit balance has dropped below the disablement threshold during the Friendly Credit Period, then the meter will disconnect upon resuming normal delivery rules.

Import: In the context of metering, the term "Import" has the same meaning as "Delivered".

Received: In the context of metering, the term "Received" refers to the quantity of a commodity that was received by the utility from the customer.

Top-up: An operation or transaction resulting in the credit balance held on a prepayment meter to be increased by use of a credit token. **NOTE -** In IEC terminology, the associated term 'vend' would normally relate to a transaction in conjunction with a vending system at a point of sale, resulting in the creation of a token that can be transported by means of a physical or virtual token carrier.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Unique Transaction Reference Number (UTRN): A numeric or alpha-numeric code used as a virtual token carrier for a payment metering system. These codes are issued by points of sale and represent an encoded and secured credit value or special action that the meter must act upon within its own application code. The functionality of payment metering tokens is outside the scope of the Smart Energy standard, however further information and examples of such tokens can be seen in IEC 62055-41.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

4 Acronyms And Abbreviations

AES	Advanced Encryption Standard
AMI	Advanced Metering Infrastructure or Advanced Metering
BOMD	Battery Operated Mirrored Device
BPL	Broadband over Power Lines
CA	Certificate Authority
CBKE	Certificate-based Key Establishment
CIN	Customer ID Number
CT	Commissioning Tool
DFT	Daily Freeze Time
ECDSA	Elliptic Curve Digital Signature Algorithm
ECMQV	Elliptic Curve Menezes-Qu-Vanstone
EMS	Energy Management System
EPID	Extended PAN Identifier
ESI	Energy Service Interface
EUI64	Extended Universal Identifier-64
GB-HRGP	HAN-ready Grouping and Protection protocol for the GB market
GPRS	General Packet Radio Service
HA	Home Automation
HAN	Home Area Network
HFT	Historical Freeze Time
HHT	Hand Held Terminal
IHD	In-Home Display
IPD	In-Premises Display (Same as IHD) or Inter-PAN Device
IVR	Interactive Voice Response
MAC	Medium Access Control (referring to protocol stack
MAC	Message Authentication Code (referring to cryptographic operation)
MDU	Multi-Dwelling Unit
MRD	Market Requirements Document
NAN	Neighborhood Area Network
PAN	Personal Area Network
PKKE	Public Key Key Establishment

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

PCT	Programmable Communicating Thermostat
PID	PAN Identifier
RFD	Reduced Functionality Device
SAS	Startup Attribute Set
SE	Smart Energy
SKKE	Symmetric Key Key Exchange
TC	Trust Center
TOU	Time of Use
UKE	Unprotected Key Establishment
UTF-8	8-bit Unicode Transformation Format Unicode Transformation Format
UTRN	Unique Transaction Reference Number
ZCL	ZigBee Cluster Library
ZDO	ZigBee Device Objects (refer to [B3])
ZDP	ZigBee Device Profile

1375
1376
1377

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

5 Profile Description

5.1 A ZigBee Smart Energy Network

The Smart Energy market requires two types of ZigBee networks for metering and energy management. These include neighborhood area networks for meters, using ZigBee for sub-metering within a home or apartment, and using ZigBee to communicate to devices within the home. Different installations and utility preferences will result in different network topologies and operation and this profile must allow for these differences. However, each of these networks will operate using the same Basic Principles to ensure interoperability.

Because of the type of data and control within the Smart Energy network, application security is a key requirement. The application will use link keys which are optional in the ZigBee and ZigBee Pro stack profiles but are required within a Smart Energy network. The Trust Center and all devices on the Smart Energy network must support the installation and use of these keys as described in the security section.

Metering networks are primarily installed by specialized service personnel, but other devices in the network may be added by home owners, or home automation professionals who may not have any ZigBee expertise. Installation concepts must be easy and uniform across Smart Energy device manufacturers.

Smart Energy networks could include both ZigBee 2007 and ZigBee 2007 Pro nodes. It is recommended the majority of the nodes in the network should be based on one stack profile or the other to get consistent performance. ZigBee Smart Energy certified products must be based upon a ZigBee Compliant Platform (ZCP). If the Smart Energy profile resides in conjunction with a private profile, the product should be ZigBee Manufacturer Specific Profile (MSP) certified and must be Smart Energy ZCP certified. This additional certification provides a reassurance that the underlying stack is behaving properly and the application is not abusive to the network.

Smart Energy networks will not interact with a consumer ZigBee Home Area Network unless a device is used to perform an “application level bridge” between the two profiles or the HA devices satisfy the Smart Energy profile security requirements. This is due to the higher security requirements on the Smart Energy network that are not required on a Home network. However, it is expected that Home Automation devices that are extended to include the Smart Energy profile can still operate in a home network.

The ZigBee Smart Energy Network makes possible networks such as the following:

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

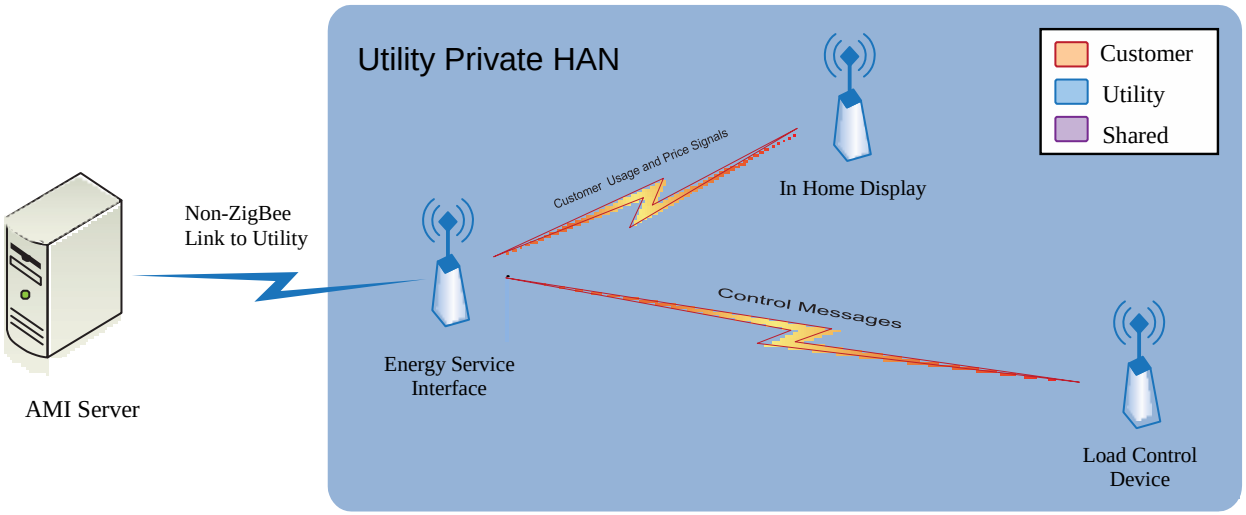


Figure 5-1 Utility Private HAN

Utility Private HAN might include an in-home display, or a load control device working in conjunction with an energy service interface, but it would not include any customer controlled devices.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

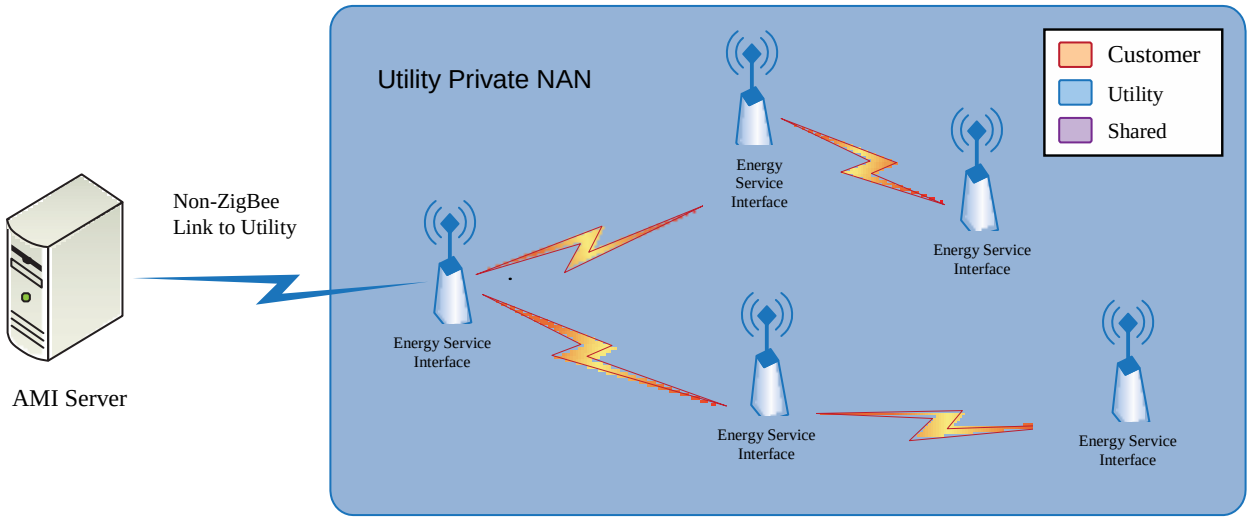


Figure 5-2 Utility Private NAN

Utility Private ZigBee network might also be used as a NAN, where ZigBee provided the primary communications for a Smart Energy deployment.

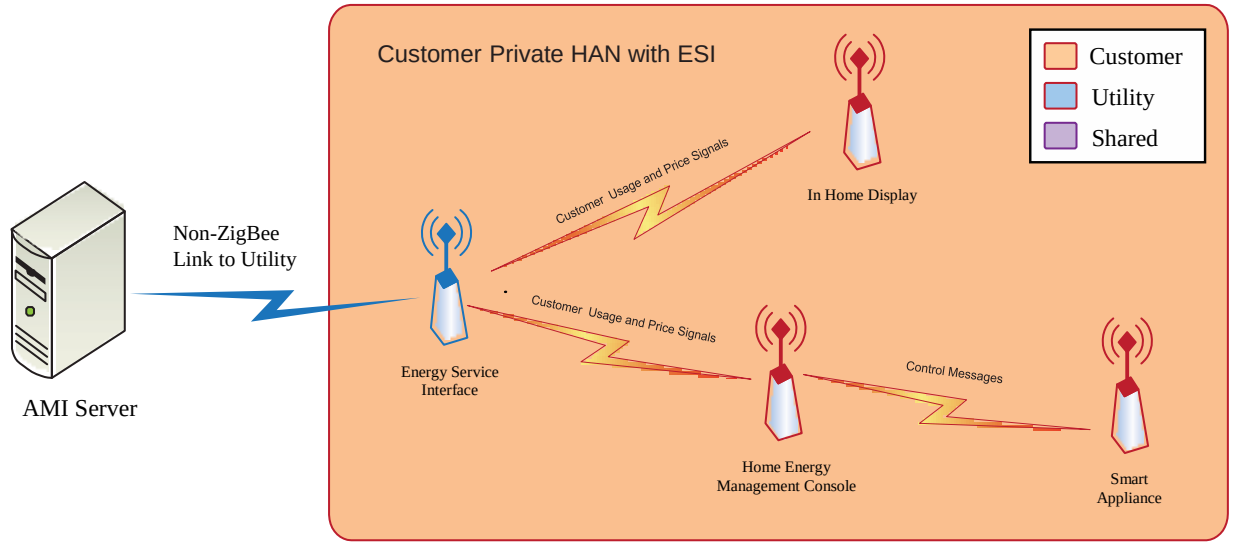


Figure 5-3 Customer Private NAN

ESI provided by utility, but limited to the role of information provider (Usage and Pricing) into a customer HAN that utilizes an Energy Management Console for conveying or controlling local devices. An example is controlling a smart appliance based upon a pricing signal.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

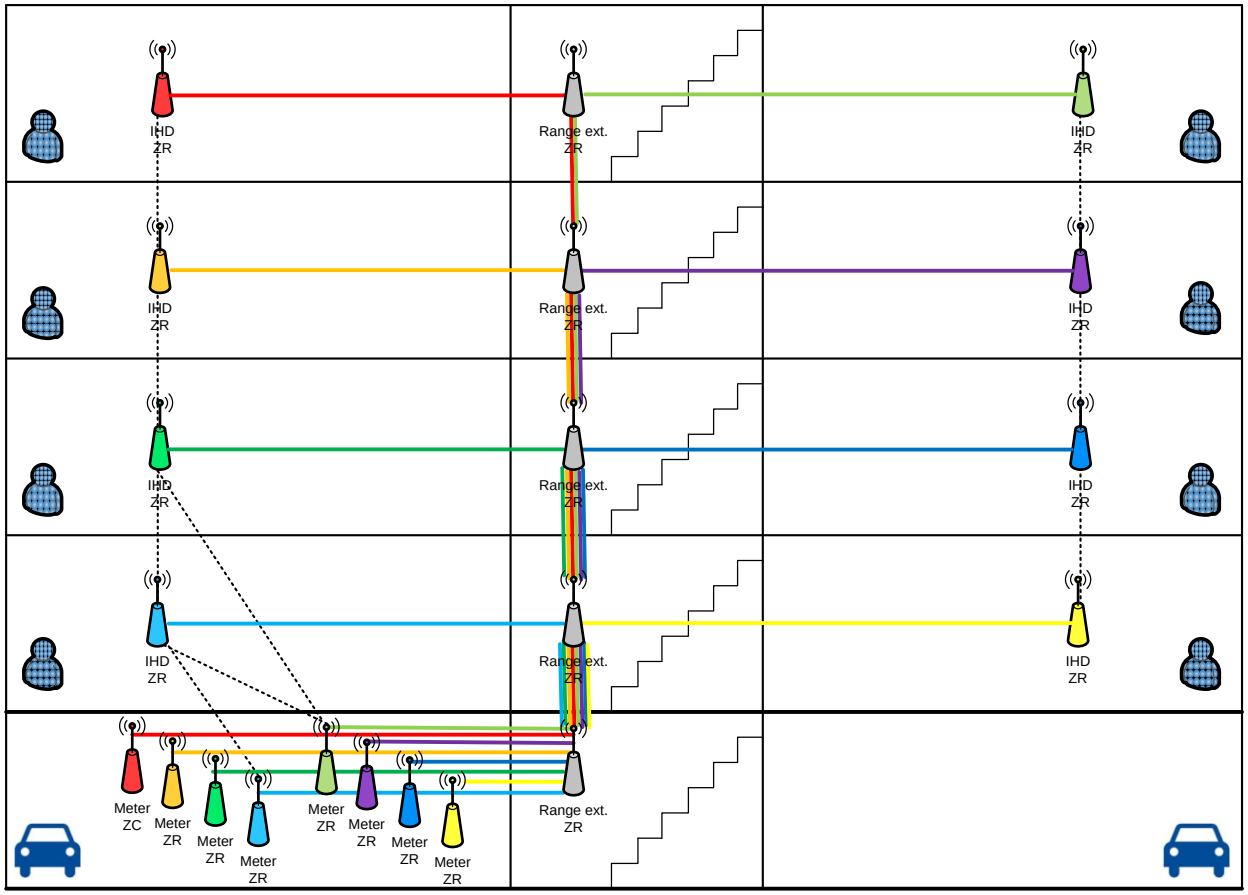


Figure 5-4 An Example of a MDU Setup

Figure 5-4 shows an example of a MDU setup, with multiple apartments in the same building (each hosting a number of ZigBee Smart Energy devices including IHDs, thermostats, range extenders etc.) and a so-called *meter room*, normally located in the basement, where electricity meter/ESI devices are installed. Range extenders are shown as well in the main stairway of the building to reinforce the mesh. The 'Federated' Trust Center (shown as Meter ZC in the diagram) provides the single back-haul connection in this example.

5.2 ZigBee Stack Profile

Products that conform to this specification shall use stack profile number 0x01 or profile 0x02, as defined in [B7]. In addition to the requirements specified in [B7], the following requirements are mandatory for this application profile.

- Support for Application link keys is required.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

- Fragmentation is required. Please refer to 5.3.8 regarding fragmentation sizes and parameter settings.

5.2.1 MAC Data Polling (NLME_Requests)

MAC Data polling is required by all sleepy end devices to operate correctly in a ZigBee Pro network. Smart Energy puts no restrictions on the frequency of MAC data polls. The choice of how frequently data polling is done will be based on individual product design considerations to reduce power consumption. However the following are a set of recommendations to ensure correct operation in the network:

- The MAC data polling rate should be dynamic based on the device's operating state. It is recommended it has at least two rates, a fast rate and a slow rate.

- The ZigBee specification only requires that parent devices buffer a single message for 7.5 seconds. This single buffer applies to all sleepy end devices. Therefore a sleepy device should poll more frequently than once per 7.5 seconds in order to be able to retrieve a buffered message.

- When the device is waiting for an active response message such as an APS acknowledgement, or a ZCL response, or participating in a multi-message protocol, it should poll at its fast rate. This fast rate is recommended to be at least once every 3 seconds.

- When the device is not actively waiting for messages it can poll at its slow rate. For example once per hour. This ensures it still has a connection with the network and with its parent.

During initial joining to the Smart Energy network, including key establishment and service discovery, it should poll at its fast rate.

5.2.2 Application Level Queries

It is expected that client devices will periodically send application level queries to servers to retrieve data. This may be done for example by thermostats querying the current price, or an in-home display to show the current reading of a meter.

Due to the fact that all ZigBee devices within the HAN utilize a shared medium for sending and receiving data it is recommended that devices do not saturate the network with frequent queries for data that does not change often. As a general rule, but not a requirement, it is recommended that devices do not initiate more than 1 query per second. This recommendation does not apply to responses generated locally due to the receipt of remote device requests. In addition, it is possible that the device may need to generate a burst of

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

traffic and exceed this recommendation. This bursting period should be very limited and followed by a period of reduced traffic respecting the above guidelines.

5.2.3 ZigBee Coordinator and Trust Center Recommendations

- In a Smart Energy based HAN network the Trust Center shall be the Coordinator (short address 0x0000).
- In a Smart Energy based HAN network the Trust Center shall be an ESI in the network.
- In a Smart Energy based NAN the backhaul point is likely to be the coordinator and trust center.

5.3 Startup Attribute Set (SAS)

In order to ensure interoperability, all ZigBee Smart Energy devices shall implement compatible Startup Attribute Sets (SAS) as defined in this specification. This does not mean that the set must be modifiable through a commissioning cluster, but that the device must internally implement these stack settings to ensure compatibility and consistent user experience. The startup set parameters described by the commissioning cluster in [B2] provide a good basis to specify a Smart Energy start up set.

Because Smart Energy Devices are likely to be preconfigured at a warehouse and installed by a technician, specific start up set values may be established by a particular utility or service area and these startup set values used in place of those below for installation. The startup set values that would be expected to be set by the installer are noted below.

5.3.1 Startup Parameters

The startup parameters and their default values are listed in Table 5-1.

Table 5-1 – Startup Parameters

Parameter	Value	Comment
Short Address	0xFFFF or installer	
E PANiD	0x0000000000000000 or installer specified.	
PAN ID	0xFFFF or installer	

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Channel Mask	All channels in frequency band.	If needed, the power transmitted by the device on channel 26 can be lowered to comply with FCC regulations.
Protocol Version	0x02 (ZigBee and later)	
Stack Profile	1 (ZigBee) or 2 (ZigBee PRO)	
Startup Control	2 (two) if un-commissioned, so it will join network by association when a join command is indicated. 0 (zero) if commissioned. Indicates that the device should consider itself a part of the network indicated by the *ExtendedPANId* attribute. In this case it will not perform any explicit join or rejoin operation.	
Trust Center Address	0x0000 (short id) installer specified Eui64.	Please note: In Smart Energy Profile 1.1 and above, only the Coordinator (0x0000) can be the SE Trust Center.
Master Key		Not used, high security is not used in this profile.
Link Key	0x00000000000000000000000000000000 00001 if the Key Establishment Cluster is being used to install a link key Installer provided if using preconfigured link keys	

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Network Key	0x00000000000000000000000000000000 00001 if no pre-installed key present	
Use Insecure Join	0x00 (False)	Flag that disables the use of insecure join as a fallback case at startup time

5.3.2 Join Parameters

The join parameters and their default values are listed in Table 5-2.

Table 5-2 – Join Parameters

Parameter	Value	Comment
ScanAttempts		At boot time or when instructed to join a network, the device should complete up to three (3) scan attempts to find a ZigBee Coordinator or Router with which to associate. If it has not been commissioned, this means that when the user presses a button or uses another methodology to join a network, it will scan all of the channels up to three times to find a network that allows joining. If it has already been commissioned, it should scan up to three times to find its original PAN to join. (ZigBee Pro devices should scan for their original extended PAN ID and ZigBee (2007) devices can only scan for their original PAN ID).
TimeBetween Scans	1 second	Determines the number of seconds between each scan attempt.
RejoinInterval	60 seconds or shorter	How quickly a device will attempt to rejoin the network if it finds itself disconnected.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

MaxRejoinInterval	15 minutes	Imposes an upper bound on the RejoinInterval parameter - this must be restarted if device is touched by human user, i.e. by a button press. This parameter is intended to throttle how often a device will scan to find its network in case the network is no longer present and therefore a scan attempt by the device would always fail (i.e., if a device finds it has lost network connectivity, it will try to rejoin the network, scanning all channels if necessary). If the scan fails to find the network, or fails to successfully rejoin, the device will wait for 15 minutes before attempting to rejoin again. To be network friendly, it would be recommended to adaptively extend this time period if successive rejoins fail. It would also be recommended the device should try a rejoin when triggered (via a control, button, etc.) and fall back to this interval if rejoins fail again.

1564

1565 ### 5.3.3 Security Parameters

1566 The security parameters and their default values are listed in Table 5-3.

1567 **Table 5-3 – Security Parameters**

Parameter	Value	Comment
SecurityTimeoutPeriod	Set by stack profile.	
TrustCenter NetworkKey	The Trust Center will pick the network key.	ZigBee Smart Energy devices shall depend on either pre-configured keys to be commissioned or the use of the Key Establishment Cluster with a pre-configured Trust Center link key to get the network key (not in the clear). ZigBee Smart Energy networks will not generally send keys in the clear.

1568

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

5.3.4 End Device Parameters

The end device parameters and their default values are listed in Table 5-4.

Table 5-4 – End Device Parameters

Parameter	Value	Comment
IndirectPollRate	Set by stack profile	This is how often a device will poll its parent for new data. It is recommended that an end device that is designed to receive data should poll its parent every 60 seconds.

5.3.5 Link Status Parameters

The link status parameters and their default values are listed in Table 5-5.

Table 5-5 – Link Status Parameters

Parameter	Value	Comment
LinkStatusPeriod	Set by stack	
RouterAgeLimit	Set by stack	
RepairThreshold	Set by stack	

5.3.6 Concentrator Parameters

The concentrator parameters and their default values are listed in Table 5-6.

Table 5-6 – Concentrator Parameters

Parameter	Value	Comment
ConcentratorFlag	Set by stack profile	Identifies the device to be a concentrator.
ConcentratorRadius	11 (eleven)	Device manufacturers that produce a concentrator product will set the max concentrator radius to this value.
ConcentratorDiscoveryTime	Set by stack profile	Identifies how often the Concentrator network layer should issue a route request command frame.

5.3.7 APS Transport Parameters

The APS transport parameters and their default values are listed in Table 5-7.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

1585 **Table 5-7 – APS Transport Parameters**

Parameter	Value	Comment
MaxFrameRetries	Set by stack profile	This determines the maximum number of retries allowed after a transmission failure.
AckWaitDuration	Set by stack profile	This is the maximum number of seconds to wait for acknowledgement of an APS frame.

1586

1587 ### 5.3.8 APS Fragmentation Parameters

1588 For fragmentation there are application settings from the APS IB that must be defined by the
1589 application profile. For Smart Energy these parameters are to be set as shown in Table 5-8.

1590 **Table 5-8 – APS Fragmentation Parameters**

Parameters	Identifier	Type	Value	Description
apsInterframe Delay	0xc9	Integer	50	Standard delay in milliseconds between sending two blocks of a fragmented transmission (see [B3] sub-clause 2.2.8.4.5)
apsMaxWindowSize	0xcd	Integer	1	Fragmentation parameter – the maximum number of unacknowledged frames that can be active at once (see [B3] sub-clause 2.2.8.4.5).

1591
1592 In addition the Maximum Incoming Transfer Size Field in the Node descriptor defines the
1593 largest ASDU that can be transferred using fragmentation. For the Smart Energy Profile the
1594 default value shall be set to 128 bytes. Maximum ASDU size allowed is specified in [B3] and
1595 dictated by solution needs and RAM capacities of the communicating devices.
1596
1597 It is highly recommended all devices first query the Node Descriptor of the device it will
1598 communicate with to determine the Maximum Incoming Transfer Size (if ASDU size is greater
1599 than 128 bytes). This will establish the largest ASDU that can be supported with fragmentation.
1600 The sending device must use a message size during fragmentation that is smaller than this value.
1601
1602 For additional information regarding the use of fragmentation in relation to the Smart Energy
1603 Tunneling cluster, please refer to Annex D.6.
1604

1605 ### 5.3.9 Binding Parameters

1606 The binding parameters and their default values are listed in Table 5-9.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

1607 **Table 5-9 – Binding Parameters**

Parameter	Value	Comment
EndDeviceBindTimeout	60 seconds	Timeout value for end device binding. End Device binding is set by the coordinator.

1608

1609 5.4 Smart Energy Profile Security

1610 To be part of a Smart Energy network, a device shall associate using one of the two
1611 association methods described below and require the use of the Key Establishment
1612 Cluster (see Annex C) for installation and updating of link keys.
1613
1614 All devices shall have the ability to retain their joining and security settings through power
1615 outages.
1616

1617 5.4.1 Joining with Preinstalled Trust Center Link Keys

1618 When using preinstalled trust center link keys, the following steps are used:

1619 **1** Trust Center link keys SHALL be installed in each device prior to joining the utility network.

1620 **2** The trust center link key for a device that is to be joined SHALL be provided to the local trust
1621 center through an out of band means as described in sub- clause 5.4.8.1 “Out of Band Pre-
1622 Configured Link Key Process”.

1623 **3** Permit joining is turned on in the network. The Trust Center enables joining by calling the
1624 NLMEPERMIT-JOINING.request primitive. Joining must be managed for an appropriate
1625 amount of time but SHALL NOT be broadcast with a time of greater than 254 seconds should
1626 not repeatedly broadcast without hearing device announcement or network administrator
1627 action. The appropriate amount of time will be dictated by the overall performance of the
1628 system and business processes driving the registration and device authorization activities. See
1629 sub-clause 5.4.1.2, “Best Practice for Coordinator Permit Joining Broadcasts”.

1630 **4** Be aware Joining has an internal time out within the ZigBee stack, therefore joining may need
1631 to be enabled multiple times during the overall Registration and device authorization process.

1632 **5** A device autonomously joining a network (i.e. without user supervision or input) may initially
1633 scan for networks to join three times in succession without pausing. After failing to
1634 successfully join a network, the device SHALL exponentially increase time between scan
1635 times, eventually performing a channel scan at a maximum rate of once per hour. The device
1636 may increase scan rate upon request from user input, such as a button push or power cycle.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

6. The device joins the network and is sent the network key encrypted with the key-transport key derived for the preinstalled trust center link key. The procedure for doing this is detailed in Annex F, also reference [B3] section 4.5.4 on key-transport keys and [B3] section 4.4.1 on frame security for the APS layer.

7. After completion of the joining procedure, the device must use the Key Establishment Cluster to establish a new link key with the trust center. The device shall use its security credentials when initiating the key establishment protocols.

8. The trust center of the network has the option of later updating the trust center link keys with devices in the network as desired by the application using the Key Establishment Cluster. Updating security keys should be an infrequent operation.

9. Once joining is completed, the list of authorized devices in the Trust Center should be updated, please refer to sub-clause 5.4.1.1, “Best Practices for Tracking Registered Devices”.

5.4.1.1 Best Practices for Tracking Registered Devices

In order to properly track Smart Energy Devices and communicate device registration status to upstream systems, Trust Centers (ESIs) should maintain a list of authorized devices. It is also recommended that Trust Centers maintain the following items for each of the registered devices:

1. Client EUI64
2. Client Installation Code
3. Registration Status
4. Time and Date Stamps
5. Supported Cryptographic Suites

With the exception of the cryptographic suite, this information is not exposed through the ZigBee network. However, the device information is expected to be used to track and understand ZigBee network connectivity.

5.4.1.2 Best Practice for Coordinator Permit Joining Broadcasts

It will be left to the coordinator / administrators of the network to determine when a network should be allowing joining. However when the network is allowing joining:

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

1. At the start of the joining period the coordinator will allow joining and broadcast a permit join message for the lesser of the permit join period or 254 seconds.

2. Every 240 seconds or whenever a device announce is received the coordinator will broadcast a permit join message for the lesser of the remaining permit join period or 254 seconds. Administrators of a network shall try to keep the amount of time devices on their networks allow joining to a minimum.

Note: *sending out a permit join message with a time of 255 (forever) is disallowed due to the risk of not being able to reliably tell devices to stop permitting joining in the future.*

5.4.2 Re-Joining a Secured Network

5.4.2.1 Rejoining Node Operation

When a device is re-joining a secured network, the following steps are used:

1. Permit joining is not required to be on in the network.

2. The device shall attempt a rejoin using the procedure detailed in [B3] Section 3.6.1.4.2 with network security. The network key and sequence number used will be the ones previously obtained from the trust center.

3. If the secured rejoin is successful, nothing more is required from the device.

4. If the secured rejoin fails, the device shall attempt a rejoin using the procedure detailed in [B3] Section 3.6.1.4.2 without network security. The re-joining device is assumed to have previously joined the network and obtained a link key using the key establishment cluster procedures. If the device does not have a link key obtained via the key establishment cluster, it cannot rejoin the network.

5. If the rejoin fails the device may attempt it again. If the device is told to leave the network it may employ the Joining using the Key Establishment Cluster procedure.

5.4.2.2 Trust Center Operation

When the trust center receives notification that a device has rejoined the network, the following steps are used:

1. If the device performed a secured rejoin the trust center is not required to take any action.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

2. If the device performed a rejoin the trust center shall determine if the device is authorized to be on the network. The trust center should send out an updated copy of the network key encrypted with the corresponding link key.

3. If the trust center determines that the device is not authorized to be on the network, it shall send an APS *Remove Device* command to the parent of the rejoining device, with the target address of the rejoining device's IEEE address. The parent will then remove that device from its child table.

Note: *The Trust Center and Router behaviors described in sections beginning at sub-clause 5.4.2.2.1 up until sub-clause 5.4.3 in this revision of this specification are provisionary and not certifiable. This text may change before reaching certifiable status in a future revision of this specification.*

5.4.2.2.1 Initiating Re-Registration

To initiate the re-registration process for a device, the Trust Center (ESI) would invalidate the Link keys for that device and subsequently cause a re- authentication / authorization to re-establish Link Keys. The processes required for this activity are:

1. The Trust Center invalidates the Link key by using the APSME-SET primitive.

2. When the Client device detects communication errors due via APS error results or by experiencing multiple re-try failures, both caused by the invalid Link Keys, it starts the processes to validate the following conditions:

   a. The Device validates its still part of the network.

   b. Route discovery processes validate communications paths are still in place.

3. If both conditions are true, the Client device attempts a secure re-join outlined in Re-joining a Secured Network and subsequently refreshes the Link Keys.

4. Re-binding of services take place (if needed).

5. Once Registration is completed, the list of authorized devices in the Trust Center should be updated, please refer to sub-clause 5.4.1.1.

5.4.2.2.2 Initiating De-Registration

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

To initiate the de-registration process for a device, which is the process of removing a previously registered device, the Trust Center (ESI) would use the following processes for this activity:

1. The Trust Center (ESI) invalidates the Link key by using the APSME-SET primitive.
2. The Trust Center (ESI) informs the Client device to leave the network by calling the NLME-LEAVE.request primitive.
3. The Trust Center (ESI) informs any Routers to remove the Client device by calling the APSME-REMOVEDEVICE.request
4. The ESI would unbind any services associated with the Client device by calling the APSME-UNBIND primitive.
5. Once de-registration is completed, the list of authorized devices in the Trust Center should be updated.

5.4.2.2.3 **Trust Center Swap-Out**

Note: *The Trust Center Swap-Out feature in this revision of this specification is provisionary and is not certifiable. This feature set may change before reaching certifiable status in a future revision of this specification.*

This section describes the requirements for swapping out a Trust Center in a Smart Energy network. In the Smart Energy Standard, an ESI should act as the coordinator and trust center of the network. In most deployments the ESI is the meter and therefore the TC. There can only be one TC in a SE network, although multiple ESIs may exist on the network. The TC (ESI) in a SE network is responsible for performing authentication and authorization. SE devices which are allowed to join the network are provisioned on the TC (ESI) from the head-end over the utility's backhaul connection.

When a TC is replaced the new device is given the extended PAN ID of the previous network, and the addresses and associated trust center link keys of all the devices from the previous network. Both the existing devices and the TC treat these keys like installation codes (unauthorized), which have limited privileges in the network. Once the devices successfully connect to the new trust center they must re-establish new TC link keys using CBKE.

Existing Smart Energy devices must be upgraded to include behavior that allows them to detect a failure to communicate with the existing trust center. When it detects this condition a device will go off in search of another network with the same extended PAN ID as the current one. If a network is found then the device will perform a first time join using the NWK rejoin and its current TC link key as the pre-configured key. If the device is able to successfully join

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

the network then it will immediately initiate CBKE to derive a new link key with the replacement TC. If that succeeds the device will identify the device's IEEE as the identity of the new trust center, and begin operating in the new network; the device shall locate any services that it may have been using.

If it is unable to join to the new network or unable to successfully negotiate CBKE, then the device will return to its previous network and continue operating. If the trust center is still unreachable at a later point in time it can perform the above steps again to attempt to find a new network.

Trust Center Swap-Out is an optional feature and is not required for ZigBee Smart Energy Certification.

5.4.2.2.3.1 SE Router Requirements

All routers in the network shall be able to identify when the trust center is no longer accessible in the network. This will be done by periodically sending an APS datagram to the Trust Center and receiving the APS acknowledgment. The APS datagram shall require encryption and acknowledgement.

After an extended period where multiple attempts have been made to contact the trust center and failed to get a response, a device would temporarily drop off the network to go in search of a network where the trust center was present. The new network may have different network parameters than the old one, but the extended PAN ID value would always be the same. Those networks that match all of the parameters of the old network will be filtered out in preference of a new network with one or more different parameters. This enables the device to find a newer instance of the existing network.

Once a new instance of the existing network has been found, the device would perform various procedures to attempt to join that network and authenticate with the new trust center device. If at any point during the attempt a failure occurred, the device may continue scanning for networks to join or return to its existing network and continue operating as it had before.

As a last resort, all devices must have a means to return to factory defaults so that they can be recomissioned. This would involve reverting back to the use of an installation code and forgetting all previous network and application parameters. This provides a means to reconnect the device to the existing network when other methods have failed, or decommission the device and join it to a new network.

5.4.2.2.3.2 Per SE Network Storage Requirements

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

TC swap-out requires the backup of data to an off-chip device. The data and storage requirements are listed in Table 5.10. Backup of the Extended PAN ID should be performed once the ESI has been commissioned or the network is formed. Backup of the TC Link Key Hash (see sub-clause 5.4.2.2.3.6) should be performed on successful completion of CBKE with the TC. TC Link Key updates from subsequent CBKE shall also be backed up. The Install Code derived TC Link Key may be backed up when the device is provisioned on to the SE network.

Table 5-10 – Per SE Network Storage Requirements

Data Description	Number of Bytes	Mandatory / Optional
Extended PAN ID	8 bytes	M
Registered device EUI64	NumberOfDevices * 8 bytes	M
Registered device Hashed TC Link Key	NumberOfDevices * 16 bytes	M
Registered device Install Code	NumberOfDevices * 16 bytes	O

5.4.2.2.3.3 Utility Requirements

It is expected that the utility is able to store backup data about each Smart Energy network in order to facilitate the TC swap-out feature. It is recommended that the list of IEEE (EUI64) addresses of devices registered in the network, and their associated installation code, always be backed up. This will help to deal with an unexpected situation due to the customer or the utility, which requires one or more HAN devices to be recommissioned.

5.4.2.2.3.4 Keep Alive Method

In order to detect the TC is no longer available all SE routers shall implement a keep-alive mechanism with the TC. The Key Establishment cluster is mandatory on all SE devices. The SE routers shall send an APS encrypted ZCL message on a periodic interval of up to a maximum of 20 minutes. The minimum polling rate should not be less than 5 minutes. Failure to receive an encrypted APS data frame (such as a read attribute response) shall indicate the TC is no longer available. If the device fails to receive 3 APS encrypted data frames in a row it shall consider the TC no longer accessible and initiate a search for it. Failure of the encryption or frame counter shall constitute a failure of the keep-alive.

5.4.2.2.3.5 Trust Center Swap-out Process

The following steps describe the Trust Center swap-out process.

Preconditions: ESI installed and PAN formed.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

1. Back up Extended PAN ID to off-chip device (mandatory).
2. SE device provisioned on the ESI and installed.
3. Back up EUI64 and Install Code to off-chip device (optional).
4. SE device performs CBKE successfully, derived TC link key and EUI64 backed up to off-chip device (mandatory).
5. Any updates to the TC Link Key must be backed up to off-chip device (mandatory).
6. SE device sends periodic APS encrypted command to the TC.
7. ESI replaced with the Extended PAN ID, list of EUI64s and hashed TC link keys restored from backed up data. (Permit joining is not required to be on in the network). The TC link keys shall be treated as install code-derived link keys and unauthorized.
8. New TC forms a new network using new network key, new short PAN ID, and backup of extended PAN ID.
9. SE device detects TC no longer available (see sub-clause 5.4.2.2.3.4).
    a. There are 4 possible cases at this point.
        i. The TC is temporarily unavailable.
        ii. The device missed a network key update.
        iii. The TC changed channels to avoid congestion.
        iv. The TC has been swapped out.
10. SE device scans for the current Extended PAN ID (the short PAN ID will probably have changed).
11. Prior to performing a rejoin the device must backup in local storage its current TC link key, state of the link key (authorized or unauthorized), network key, and associated NWK and APS frame counters.
12. If the TC sends a Transport Key message encrypted using the device's existing TC link key and the device is able to successfully decrypt and rejoin the network, no further operations are necessary. The device can resume all normal operations.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

13 If the TC sends a Transport Key message encrypted using a 128-bit AES-MMO hash of the TC link key, then the device shall ignore the frame counter check and accept the new network key. It will also record the source IEEE address of the sending device as the new TC address. It shall mark the hashed TC link key as not authorized and treat the link key as an installation code. It must now perform Key Establishment to fully authenticate itself in the network.

14 If the key establishment is NOT successful the device may try again immediately. Otherwise it shall leave that network. It can continue scanning for additional networks to rejoin, or restore the values of its previous network and resume normal operation.

15 If the Key Establishment is successful then the device can resume normal operation. It may discard the backup of security data from the old network. The TC must backup hash of the new TC Link Key for this device to an off-chip device.

16 If after attempting rejoin with all discovered PANs fails then the device shall fall back to the existing PAN.

5.4.2.2.3.6 Link Key Hash

In order to protect the data that is being backed up, a hash on the TC link key will be performed and that will be the key stored externally. It is highly recommended that the actual link key used for operational networks never be transported out of the ESI. Using this method if the backup data for the TC is compromised then it cannot be used to compromise existing ZigBee network communications.

The hashed key shall be created by performing a 128-bit AES-MMO hash on the 128-bit key data. The following is a test vector for the hash:

Table 5-11 – Example Hash of the TC Link Key

TC Link Key	C0C1C2C3C4C5C6C7C8C9CACBCCCDCECF
Hashed TC Link Key	A7977E88BC0B61E8210827109A228F2D

5.4.2.2.3.7 Trust Center (ESI/Meter)

Dependencies

The ESI shall support backup and restore of data (including TC link keys) to an off-chip device.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Routers shall detect the TC is no longer available by sending an APS encrypted command and receiving the APS acknowledgement with a maximum periodic interval of 30[1] minutes.

Table 5-12 – Parameters of Trust Center Swap-Out

Name	Type	Range	Default	Mandatory / Optional
TC Keep-Alive	Unsigned 8-bit integer	0x01 - 0x1E	0x14	M

5.4.3 Devices Leaving the Network

Upon receipt of an APS update device command indicating a device has left the network the trust center shall not remove the trust center link key assigned to that device. This is to prevent a device on the network performing a denial of service attack by spoofing the MAC address of another node and issuing a false ZigBee *Network Leave* command. Devices should be removed from Trust Center authorization and trust center link key lists via out of band methods, i.e. secure meter back haul or secure IP interface.

Devices should follow the guidelines for stale keys described in 5.4.5.

5.4.4 Updating the Network Key

Periodically the trust center shall update the network key. This allows the trust center to phase out a previous instance of the network key so that devices that are no longer on the network will not be able to perform a secure rejoin. Those devices must then perform a rejoin, which allows the trust center to authorize whether or not they are allowed to be on the network.

When the trust center wishes to update the network key it will broadcast the network key to all devices in the network. All devices receiving the key update will store but will not start using the new key.

It is assumed that routers will receive the network key update sent by the Trust Center. Sleepy end devices are unlikely to get the network key update sent by the Trust Center unless the device polls frequently.

After sending an updated network key, the trust center shall wait a minimum of nwkNetworkBroadcastDeliveryTime before sending the switch key message. Devices that miss the key switch broadcast message will implicitly switch when they receive any network message that is encrypted using the new key sequence number.

[1] CCB 1746

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Once the network has started using the new key, any device that has missed the key update message will not be able to communicate on the network. Those devices that missed the key update must follow the Re-joining a Secured Network procedure.

5.4.5 Updating the Link Key

Periodically the trust center may update the link key associated with a particular device. This allows the trust center to phase out the existing key and refresh it with a new key. The trust center can decide on its own what the policy is for how long a link key may be used and how often it should be updated.

Trust Center link keys are used for sending application messages as well as stack commands. Therefore a trust center cannot simply delete a link key that it wants to update. The trust center must accept and or send encrypted APS commands to or from a device even if it has retired that link key from encryption of application data messages. This is especially necessary for sleeping end devices, which may not have the current network key and need to use their link key to obtain an updated copy during a rejoin.

When the trust center deems that a particular link key should no longer be used, it shall mark the key as stale. A stale key shall not be used to send data messages. Devices that receive a message using a stale key should discard the message and shall not send an APS acknowledgement to the sender.

Devices shall accept and process APS commands that are encrypted with a stale key.

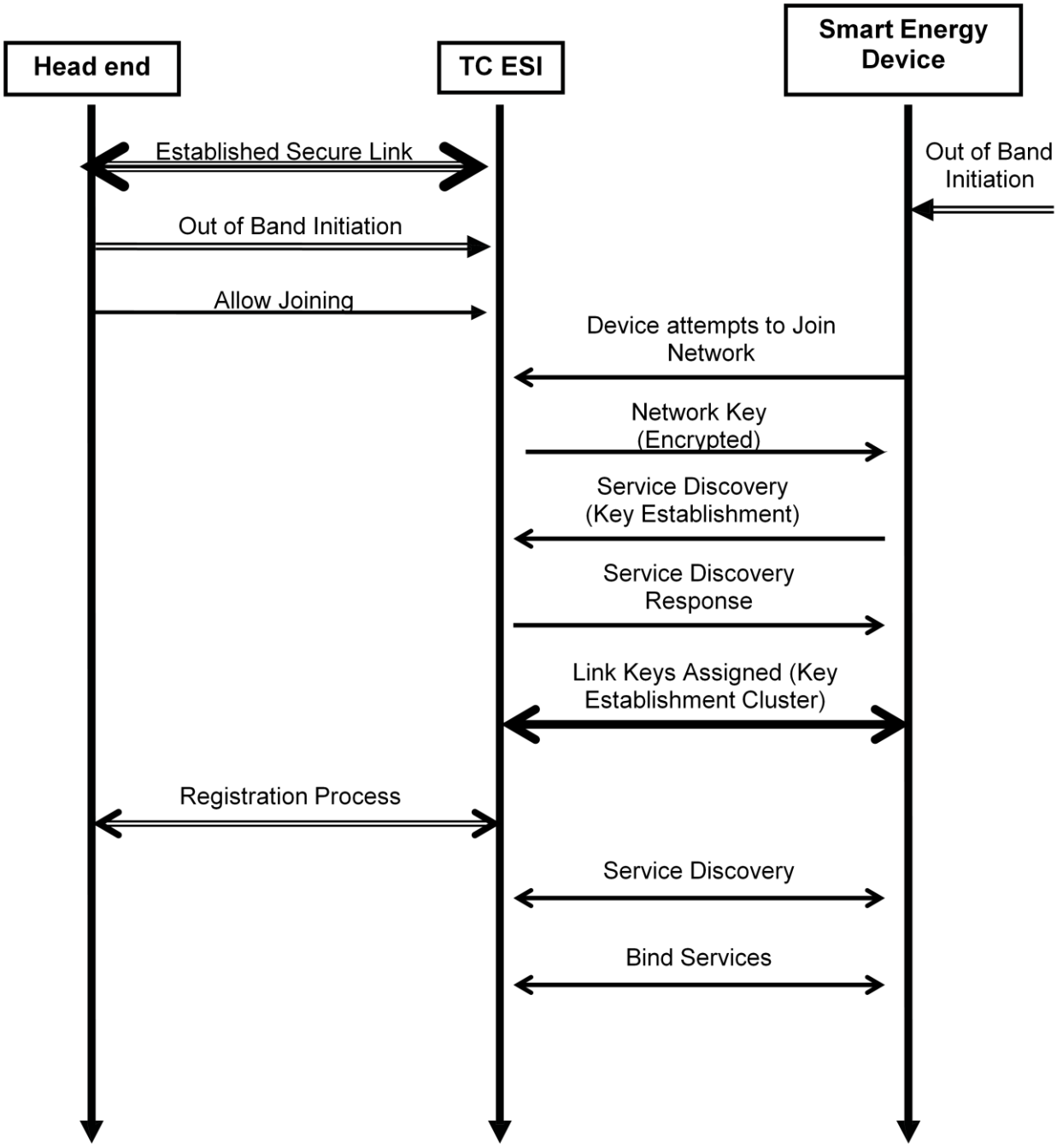
When the trust center receives a message encrypted with a stale link key, it shall initiate the key establishment procedure to negotiate a new link key. Upon successful establishment of the new link key with the device, the device shall clear the stale indicator for that key.

Devices that are not acting as the trust center may utilize their own policy for retiring and updating application link keys with other devices that are not the trust center. Those devices are not required to keep around retired keys and therefore may delete them prior to establishing an updated link key using the Key Establishment cluster.

5.4.5.1 Network Joining and Registration Diagram

Figure 5-5 depicts an example of a successful network startup and certificate exchange (with pre-established link keys). Please refer to Annex C for further discussions on communication exchanges and key support.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.



1946

1947 **Figure 5-5 Successful Join and Registration**

1948 *Please note:* After joining the network and acquiring a Network Key, the Smart Energy End
1949 Device shall initiate the Service Discovery process to locate the Key Establishment Cluster. As
1950 recommended best practice, the TC ESI should support a fault-tolerant behavior by initiating

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Key Establishment Cluster service discovery process whenever it detects the Smart Energy End Device fails to do so.

After Joining and after Key Establishment:

- A device supporting MDUs shall request MDU Pairing Information from the Trust Center (ESI). If MDU Pairing Information is returned, the device should limit service discovery to those devices identified in the returned information. If the device has joined a 'normal' Smart Energy HAN, no Pairing Information will be returned and the device should perform service discovery with all devices on the network. Refer to 5.6 and Annex D.13 for further details.
- Client SHALL perform service discovery.
- Sleepy devices SHALL perform "get" requests for data they wish to receive and SHOULD NOT expect to receive unsolicited messages.
- If a Client wishes to receive unsolicited messages, Client SHALL follow with attempt(s) to ZDO Bind Request. A Client does not have to support a binding table.
- If Server does not support binding, Server SHALL perform service discovery and register those devices for unsolicited messages (whether or not they want the messages).
- For backward compatibility, Server SHOULD perform service discovery and register those devices for unsolicited messages (whether or not they want the messages).

5.4.6 Cluster Usage of Security Keys

The SE Profile utilizes a higher level of security on the network but not all clusters need to utilize Application Link keys. All clusters are required to use network layer encryption using the network key. Table 5-13 identifies the security keys utilized by each cluster:

Table 5-13 – Security Key Assignments per Cluster

Functional Domain	Cluster Name	Link Key Required
General	Basic	No
General	Identify	No
General	Alarms	No
General	Time	Yes
General	Commissioning	Yes

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

General	Power Configuration	No
General	Key Establishment	No
Smart Energy	Price	Yes
Smart Energy	Demand Response and Load Control	Yes
Smart Energy	Metering	Yes
General	Over the air Bootload Cluster	Yes
Smart Energy	Messaging	Yes
Smart Energy	Tunneling	Yes
Smart Energy	Prepayment	Yes
Smart Energy	Calendar	Yes
Smart Energy	Device Management	Yes
Smart Energy	Events	Yes
Smart Energy	MDU Pairing	Yes
Smart Energy	Energy Management	Yes

Once a Registered SE device has an Application Link Key established with the ESI, it may also establish Application Link Keys with any other device on the SE Network. This is accomplished by using the ZigBee service and device discovery process (employing the Network Key). Note that, in an MDU network, the service and device discovery process can only commence once MDU Pairing information has been acquired, and will then be limited to devices within the same 'virtual HAN'. Regardless of the communication paths, all SE applications shall use and validate the Security key usage as listed in Table 5-13. If link key encryption is NOT used but required, the receiving device shall generate a ZCL Default Response, employing the Network Key, with a FAILURE (0x01) status code.

It is permissible for a device to initiate a ZCL exchange using an application link key even when not required. If a device receives a message with link key security even though it is not required as per Table 5-13, it shall accept the message. Additionally, if a response is sent then it shall use link key encryption.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Unless stated otherwise, any ZCL clusters added to a Smart Energy endpoint shall be APS encrypted. ZCL clusters without APS encryption shall be located on another (non-Smart Energy) endpoint. The same restrictions shall also apply to manufacturer-specific clusters.[2]

5.4.7 Key Establishment Related Security Policies

The following are the policies relating to Key Establishment that are recommended for Smart Energy networks.

5.4.7.1 Joining

If the device does not need to perform discovery queries or other non-secure operations after it joins an SE network and receives the Network Key, it should immediately initiate Key Establishment with the Trust Center to obtain a new Trust Center Link Key.

If Key Establishment fails with a result of UNKNOWN_ISSUER the device shall leave the network. A device that does not initiate Key Establishment with the Trust Center within a reasonable period of time MAY be told to leave depending on the network operator's policy. A maximum period of 20 minutes is recommended.

Upon successful negotiation of a new Trust Center Link Key the device may communicate using clusters that require APS security.

5.4.7.2 Trust Center

The Trust Center shall keep track of whether a particular device has negotiated a CBKE Trust Center Link Key, or whether only a preconfigured Trust Center Link Key exists. The Trust Center shall not use the preconfigured link key to send encrypted APS Data messages to the device. The Trust Center shall discard any APS encrypted APS Data messages that use the preconfigured link key, and it shall not send APS Acks for those messages.

The Trust Center shall accept and send APS Data messages that do not use APS Encryption to a device that has not negotiated a CBKE Trust Center Link key provided that the security usage for that cluster allows using only Network layer security (encrypted with the Network Key). See sub-clause 5.4.6, "Cluster Usage of Security Keys".

The Trust Center is required to be a Smart Energy device. It is required to support Key Establishment server on at least one endpoint, though it may support it on more than one endpoint. These endpoints shall be considered to all refer to the same logical ZigBee device type, in other words the Trust Center. Any negotiation or establishment of a link key on one endpoint applies globally to the Trust Center as a device and is not specific to an endpoint.

[2] CCB 1501

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

The Trust Center shall have a means of adding and removing keys of specific devices that are part of the Smart Energy network. The specific means of doing this is outside the scope of this document.

5.4.7.3 During Joining

Normal operation of a device in a Smart Energy network requires use of a preconfigured link key, established by using the Installation Code (refer to sub- clause 5.4.6), to join a ZigBee Pro network. After joining the network a device is required to initiate key establishment using ECMQV key agreement with the Trust Center, to obtain a new link key authorized for use in application messages.

Prior to updating the preconfigured link key using key establishment, the Trust Center shall not allow Smart Energy messages that require APS encryption. Although the node has a link key, that node has not been authenticated and thus the key's use is not authorized for application messages. Its use is still required for certain stack messages (e.g., the APS Command Update Device) and must be accepted by the trust center.

In order to perform key establishment the device must discover an endpoint on the Trust Center that supports the Key Establishment server cluster. The joining device shall perform a ZDO Match Descriptor Request to determine what endpoint to use. This request shall be unicast to the Trust Center's short address of 0x0000.

When a reply is received, it may contain multiple endpoints that indicate support for the Key Establishment server. The joining device may use any endpoint to perform key establishment. Link keys established using key establishment are global to the Trust Center device and are not specific to a particular endpoint.

Once a node has been authenticated by the Trust Center and obtained an authorized link key using key establishment, it may communicate with the Trust Center using APS layer security. The Trust Center should accept valid APS encrypted message using that new link key. At this point the joining device can communicate to the Trust Center as a Smart Energy device.

5.4.7.4 After Joining

After a node has joined, been authenticated using key establishment, and obtained an authorized link key, it may need to communicate with other nodes on the network using APS layer encryption.

Rather than use key establishment with each node on the network, it would be advantageous to leverage the Trust Center to broker trust with other devices on the network. If two nodes have both obtained link keys with the Trust Center using key establishment, then they both trust the Trust Center. Both nodes will use the Trust Center to request a link key with each other. The

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

trust center will respond to each node individually, sending a randomly generated link key. Each message will be encrypted using the individual nodes' link keys. The Trust Center would not send a link key to either node if one of the nodes has not authenticated using key establishment.

The originating node would start this process by sending a bind request command with APS ack to the Key Establishment cluster of the destination device. If a bind confirm is received with a status of success, the initiating device will perform a request key of the trust center (for an application link key using the EUI of the other device in the pair). The trust center will then send a link key to each device using the key transport. If the bind confirm is received with a status other than success, the request key should not be sent to the trust center.

This functionality is optional, however support of this is required for ESI devices acting as trust centers. All devices sending the request key command and the trust center should have a timeout of 5 seconds.

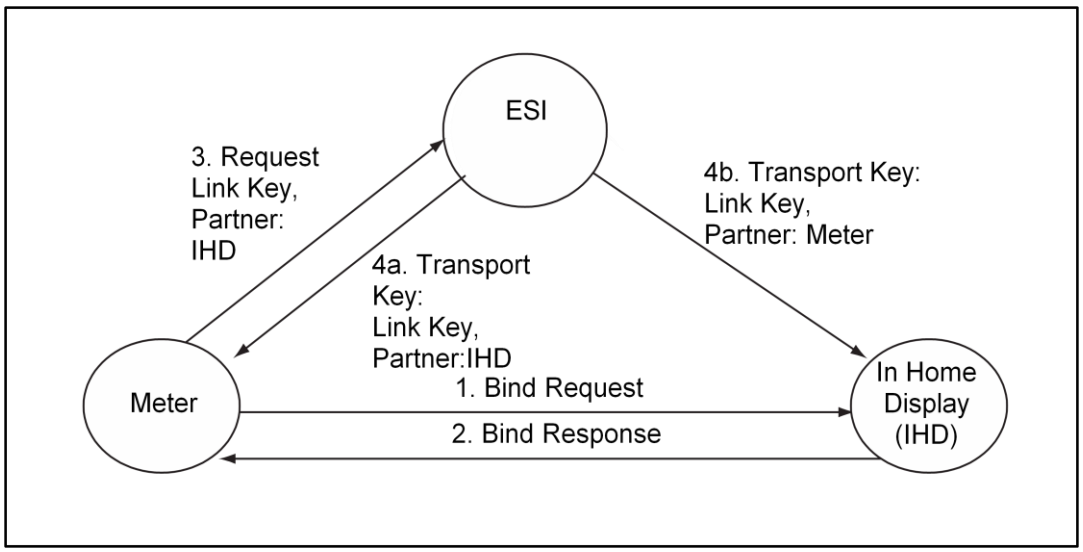


Figure 5-6 Node Communication with Other Nodes on the Network Using APS Layer Encryption

The advantages of using the stack primitives to request keys rather than key establishment are that devices can forego the expensive ECC operations. Small microprocessors have extremely limited resources and requiring full key establishment with all devices where link keys are required is overly burdensome. In addition, ESIs may have other security policies in place (such as node blacklists or certificate revocation lists) that individual nodes do not have knowledge of, or have the resources to keep track of.

Nodes that are not the trust center would not be allowed to initiate key establishment with another device that is not the Trust Center. If a device receives an Initiate Key Establishment Request from a device that is not the Trust Center, and it is not the Trust Center, it shall

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

terminate the key establishment immediately with a status of NO_RESOURCES. This ensures that the ESI authenticates all devices with key establishment after joining, and limits the use of key establishment in the network.

Other ESI devices on the network that are not the trust center would have to go through the same procedure as above, contacting the ESI trust center, in order to send/receive messages that require APS layer encryption with another node.

5.4.8 Security Best Practices

5.4.8.1 Out of Band Pre-Configured Link Key Process

This section describes the out of band process for establishing pre-configured Trust Center link keys, the format of the Installation Code required, and the hashing function used to derive the pre-configured link key from the Installation Code.

As portrayed in Figure 5-7, during the manufacturing process a random Installation Code is created for each of the Smart Energy devices. This Installation Code is provided for the device in a manufacturer-specific way (labeling, etc.) and referenced to during installation. The space of installation codes should possess the same randomness properties as a key space. Knowing a set of installation codes should not yield any knowledge of another installation code; and each installation code should be equally probable. The associated Pre-configured Link Key is derived using the hashing function described below and programmed in the device.

Step 1: An Installation Code is created and made available

Step 2: The Pre-configured Link Key is derived from the Installation Code using the Matyas-Meyer-Oseas hash function

Step 3: The Pre-configured Link Key is configured in the device

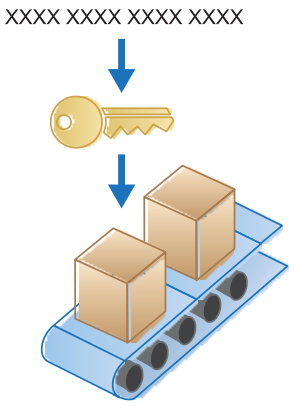


Figure 5-7 Smart Energy Device Installation Code Process

As portrayed in Figure 5-8, during the installation process the initial Trust Center Link Key is derived from the Installation Code and sent via an out of band communication channel to the

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Trust center (ESI). The Trust center uses this Key as the Trust Center Link Key to subsequently configure the Network Key of the associating device.

Step 1: The Installation Code is sent out of band

Step 2: The Pre-configured Link Key is derived from the Installation Code using the Matyas-Meyer-Oseas hash function

Step 3: The Pre-configured Link Key is sent to the Trust Center using the AMI network

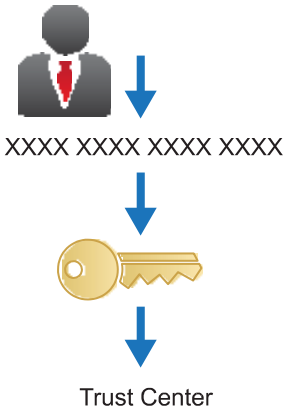


Figure 5-8 Installation Code Use with the Trust Center

5.4.8.1.1 Installation Code Format

The Installation Code consists of a 48, 64, 96, or 128 bit number and a 16 bit CRC (using CCITT CRC standard polynomial X16 + X12 + X5 + 1). When printed or displayed, Installation Codes are represented as multiple groups of 4 hexadecimal digits.

48 Bit example:

Installation Code of "83FE D340 7A93 2B70"

Where values 0x83, 0x FE, 0xD3, 0x40, 0x 7A, and 0x93 are used to calculate the CRC16 with the result returning 0x702B.

Note: *The Octet order of the CRC code in the printed Installation code is Least Significant Octet followed by Most Significant Octet, giving the printed result of "2B70".*

64 Bit example:

Installation Code of "83FE D340 7A93 9738 C552"

Where values 0x83, 0x FE, 0xD3, 0x40, 0x 7A, 0x93, 0x 97, and 0x38 are used to calculate the CRC16 with the result returning 0x52C5.

96 Bit example:

Installation Code of "83FE D340 7A93 9723 A5C6 39FF 4C12"

Where values 0x83, 0xFE, 0xD3, 0x40, 0x7A, 0x93, 0x97, 0x23, 0xA5, 0xC6, 0x39 and 0xFF are used to calculate the CRC16 with the result returning 0x124C.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

128 Bit example:
Installation Code of “83FE D340 7A93 9723 A5C6 39B2 6916 D505 C3B5”
Where values 0x83, 0xFE, 0xD3, 0x40, 0x7A, 0x93, 0x97, 0x23, 0xA5, 0xC6, 0x39, 0xB2, 0x69, 0x16, 0xD5, and 0x05 are used to calculate the CRC16 with the result returning 0xB5C3.

5.4.8.1.1.1 CRC Algorithm Information

As stated earlier, the Installation Code CRC calculation is based upon the CRC 16-CCITT algorithm and uses the following parameters:

Length: 16
Polynomial: x16 + x12 + x5 + 1 (0x1021)
Initialization method: Direct
Initialization value: 0xFFFF
Final XOR value: 0xFFFF
Reflected In: True
Reflected Out: True

Open source implementations of the CRC 16-CCITT algorithm are available on the internet at sites like SourceForge and others. The source code is also available for download from the ZigBee document management system [B6].

5.4.8.1.2 **Hashing Function**

An AES-128 key is derived from the Installation Code using the Matyas-Meyer- Oseas (MMO) hash function (specified in Annex B.6 in ZigBee Document 053474r17, The ZigBee Specification, ZigBee Technical Steering Committee (TSC) with a digest size (hashlen) equal to 128 bits).

Installation Code examples:

- MMO hash applied to the Installation Code “83FE D340 7A93” produces the key “CD4FA064773F46941EC986C09963D1A8”.

Note: *Least significant byte is 0x83 and Most significant byte is 0x93.*

- MMO hash applied to the Installation Code “83FE D340 7A93 9738” produces the key “A833A77434F3BFBD7A7AB97942149287”.

Note: *Least significant byte is 0x83 and Most significant byte is 0x38.*

- MMO hash applied to the Installation Code “83FE D340 7A93 9723 A5C6 39FF” produces the key “58C1828CF7F1C3FE29E7B1024AD84BFA”.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Note: *Least significant byte is 0x83 and Most significant byte is 0xFF.*

- MMO hash applied to the Installation Code "83FE D340 7A93 9723 A5C6 39B2 6916 D505" produces the key "66B6900981E1EE3CA4206B6B861C02BB".

Note: *Least significant byte is 0x83 and Most significant byte is 0x05.*

5.4.8.1.2.1 MMO Hash Code Example

Open source implementations of the MMO Hash based on the Rijndael implementation are available on the internet at sites like SourceForge and others. The source code is also available for download from the ZigBee document management system [B6].

5.4.8.2 Multiple Security Credentials

The Key Establishment cluster supports multiple cryptographic suites and, as a result, a device's security credentials (e.g. certificates) are specific to the cryptographic suite. All clients and servers SHALL implement and support all crypto-suites. Markets and/or specific deployments may choose to only use certain crypto-suites at runtime based on their policies.

However, the trust center dictates security policy for the network and as such may require a higher level of security by using certain cryptographic suites as determined by the network owner. Therefore it is possible that trust center devices deployed into new regions with no existing Smart Energy may utilize only newer cryptographic suites that have a higher level of cryptographic strength.

5.5 Commissioning

Many, if not all of the devices described in this document, will require some form of commissioning, even if the user or installer doesn't see it. This is because, for example, a load control device needs to be bound to some sort of control device in order to perform its function and, even if the required initializations are done at the factory before the device is installed, the required operations are virtually the same as is the outcome.

The ZigBee Alliance has recognized the importance of commissioning and, in particular, the importance of specifications for network and stack commissioning in a multi-vendor environment. Thus, network and stack commissioning procedures are being designed outside the context of any particular profile, where possible, and grouped under the auspices of the Commissioning Tools Task Group (CTTG). This task group is developing a commissioning framework specification [B2].

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

5.5.1 Forming the Network (Start-up Sequence)

Smart Energy devices must form their own network or join an existing network. The commissioning framework [B2] discusses some of the relevant issues in this procedure.

It is intended that an installer of a Smart Energy device knows if the device is forming a network or joining an existing network.

If a device is forming a network there is no user interaction required since the form process can be completed by the device. However there should be some indication to the user or installer that the network has formed properly. The indication can be implemented in a number of ways including blinking indicator lights, colored indicator lights, arrays of indicator lights, text displays, graphic displays, audible indicators such as buzzers and speakers, or through separate means.

If a device is joining an existing network, it will join the network using the processes outlined in sub-clause 5.4. Permit joining will have been turned on due to either installer action or some backchannel mechanism because of user or installer action. It is recommended there be some indication to the user that the device has joined the network successfully. The indication can be implemented in a number of ways including blinking indicator lights, colored indicator lights, arrays of indicator lights, text displays, graphic displays, audible indicators such as buzzers and speakers, etc.

5.5.2 Support for Commissioning Modes

Three different commissioning modes are discussed in [B2]. They are denoted A, E and S-mode.

As discussed above, Smart Energy devices will either automatically form or join a network based on the processes outlined in sub-clause 5.4.

The pre-installation of start up parameters could be done at manufacturing (which is defined as A mode), by an installer tool at the dispatching warehouse, or on site (which would then be S mode). Devices that support this pre-installation must document the methods used for this preinstallation of parameters to accomplish this process.

Those devices that will join an existing network must support button pushes or simple documented user interfaces to initiate the joining process. This is in support of E mode commissioning.

5.5.3 Commissioning Documentation Best Practices

To ensure a uniform user experience when commissioning Smart Energy devices, all ZigBee Smart Energy devices are required to provide documentation with their product that explains how

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

to perform device commissioning in using a common language set, i.e., “form network”, “join network”, etc. Please refer to [B2] for further guidance using installation tools and procedures.

5.5.4 Commissioning Procedure for Different Network Types

Depending on the type of network being installed, the commissioning procedures may be slightly different. To ensure interoperability even within these different methods the specific steps are detailed here.

5.5.4.1 Commissioning for Neighborhood Area Network or Sub-metering

Under a neighborhood area network, other meters such as gas or water meters may join electric meters that form a backbone of the network. The process of joining the network is separate from the process for device binding where the device billing information is configured for a particular dwelling unit. It may be desirable to allow the meter to join an adjacent dwelling unit from a network standpoint to ensure proper connectivity. The application level will handle the configuration of the billing information later.

1. There are two methods for joining such a device onto an existing network:
   a. The device is commissioned using a tool with the necessary network and security start up parameters to allow it to rejoin the network as a new device. The device can rejoin any device in the network since it has all the network information.

   b. The network has permit joining turned on by an external tool and the device joins this network and undergoes joining and authentication as any newly joined device.

2. Once joined and authenticated by the security requirements of the existing network, the device is now a member of the neighborhood area network.

3. At the application level, the particular device ID is associated with a particular dwelling unit for billing purposes. This information may be associated at the backend database where the data is collected, or may be sent to the device so it is aware of its association. Note that under this method, devices may route data through devices in adjacent dwelling units that are part of the neighborhood area network.

5.5.4.2 Commissioning for Home Area Network

Under a home area network, the network consists of devices in a particular dwelling unit with one or more co-located metering devices or ESI that provides connectivity to the utility network. Under this scenario, the device within the home may be installed by a trained installer or by a homeowner. The following steps are completed:

1. The Smart Energy network must be informed of the device that is to be joined. This is done through an out of band means which could include a web login, phone call to a service center, or handheld tool. Using this methodology the existing network is made aware of the device ID

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

and security information appropriate for the device (per the Key Establishment Cluster described in Annex C).

2. The Smart Energy network is put into permit joining ON for a period of time.

3. The installer/homeowner is prompted to press a button or complete a menu sequence that tells the device to attempt to join a network.

4. The device joins the network and is authenticated using the appropriate security mechanisms per the Key Establishment Cluster.

5. An indicator is provided for the installer/homeowner indicating the device has joined a network and authenticated properly or provides information about improper authentication.

6. The device can now operate normally on the network.

5.5.5 ZigBee Smart Energy Joining, Service Discovery, and Device Binding Requirements

Commissioning of a device into a ZigBee Smart Energy network should be easy, reliable, and deterministic. Ideally, a new device could be installed by the home owner or installer communicating the device install code out of band to the coordinator/trust center and then simply powering up the device or manually putting the device into a commissioning (auto-join) state. The device should automatically handle all the steps needed to discover and join the correct PAN and establish relationships with other devices in the HAN without user intervention. As network or HAN conditions change, the devices should be able to adapt automatically without user intervention. ZigBee Smart Energy networks are supposed to last for decades, but once commissioned, devices should require no user interaction in order to remain part of the ZigBee PAN.

Devices that are configured with a Startup Parameter of two (un-commissioned) should automatically begin or make easily available a way to go to Auto-Joining State as described below. (See sub-clause 5.3.1 for the SE Profile Startup Parameter set.)

5.5.5.1 PAN Auto-Joining State

1. When auto-joining state is initiated, a device shall periodically scan all startup set channels for networks that are allowing joining. (See sub-clause 5.3.1 for startup set channel description). A recommended periodic schedule would be:
   a. Immediately when auto-joining state is initiated.
   b. If auto-joining state fails, retry once a minute for the next 15 minutes, jittered by +/- 15 seconds.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

c If those joining states fail, then retry to join once an hour jittered +/- 30 minutes.

2 To find prospective networks to join, the joining node shall send Beacon Request packets on each channel, dwelling on each channel as specified by the ZigBee PRO specification beacon response window.

3 When a beacon is heard and it has the "Permit Joining" bit set, the device shall attempt to join that PAN. It is up to the implementation of the device to decide if it wants to survey all channels and build a list of joinable PANs before attempting a join procedure, or if it should attempt to join each PAN on a beacon-by-beacon basis. The device shall use its preconfigured link key (derived from a hash of the installation code) to join the targeted SE PAN. Exchanging keys in the clear or with well known preconfigured link keys is not allowed.

4 If the device joins the network but receives a network key that it cannot decrypt, then it has likely joined an incorrect PAN and should back out and try the next joinable PAN. This situation happens most often when the out of band mechanism to communicate the installation key is flawed, or when more than one PAN is allowing joining. It is permissible to try and join the same network again, but not recommended that it be done more than three times in succession. It is expressly not allowed that a device repeat this step more than ten times without backing off to step two and scanning for other networks to join.

5 After the device joins the PAN and is granted ZigBee network key, it must perform service discovery to find a ZigBee Key Establishment cluster server, then perform ZigBee key establishment in order to get an APS layer link key.

6 If this key establishment fails, it is likely that one side of the exchange is configured with an invalid certificate or with no certificate at all. It is permissible to retry this step multiple times in succession, but it is expressly not allowed that a device repeat this step more than ten times in succession without pausing for a minimum of least fifteen minutes. Since the device was able to get a network key from the Trust Center, the device must have found the correct PAN to join, so there is no need to leave the network. A device that does not initiate Key Establishment with the Trust Center within a reasonable period of time MAY be told to leave depending on the network operator's policy. A maximum period of 20 minutes is recommended.

7 Once key establishment succeeds, the device has joined the correct PAN and shall never leave the PAN without direction from another device in the network (typically an APS *Remove Device* command from an ESI or ZigBee Network Manager) or direction from the user via the device user interface. Example user interfaces could be a text menu or a simple button push sequence. It is strongly recommended that the user interface procedure to get a device to leave the PAN be explicit and difficult to trigger accidentally. Leave commands received over the air should only be followed if the command is an APS encrypted APS *Remove* command. Network layer leave commands should be ignored unless the device is an end device, and the network leave command originated from the parent device.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

8 A device that leaves a ZigBee network shall discard its network settings and link key, and revert to its install code, and wait for user input to return or automatically return to auto-join state step one. The device will require the out- of-band registration process to join a new network.

5.5.5.2 Service Discovery State:

1 After successfully performing key establishment, devices supporting MDU shall perform service discovery for the MDU Pairing cluster first and, if found, request pairing information before continuing discovery for other services. If the MDU Pairing cluster is NOT located on the Trust Center (ESI), the device providing the cluster MUST be available on the HAN before other devices are allowed to join. If MDU Pairing Information is returned, the device should limit service discovery to those devices identified in the returned information. If the device has joined a 'normal' Smart Energy HAN, no Pairing Information will be returned and the device should perform service discovery with all devices on the network. Refer to 5.6 and Annex D.13 for further details.

2 The device should use ZigBee Service Discovery mechanisms to discover other devices on the network that have services that match with the device's. This would apply to ZigBee Smart Energy clusters that support asynchronous event commands, like DRLC, Messaging, and Price clusters. For example, a load control device would use ZigBee service discovery to find ESIs that support the load control cluster server. (See sub-clause 5.4.5.1 for more details.).

3 Where a network supports multiple commodity types, the service discovery mechanism may find multiple instances of certain clusters. In this case, the device should refer to the *CommodityType* attribute within any Price clusters found, or the *MeteringDeviceType* attribute within any Metering cluster servers found, in order to determine the appropriate endpoint(s). Price clusters must always be used where multiple commodities are supported; all clusters related to the same commodity should be grouped on the same endpoint. If an endpoint supporting the required commodity is not found, a device shall not use clusters identified as being for another commodity; the device shall repeat the discovery mechanism at regular intervals until such time as the correct clusters become available (see point 6 below for suggested intervals). Clusters for multiple commodities shall not be mixed on the same endpoint (see section 6.3 for further details).

4 When a matching service is discovered, the device shall use ZigBee device bind mechanisms to send a binding request to the matching device endpoint. It is possible that more than one device with matching services will be discovered. If the device is not an ESI and the ESI are the matching device(s), the device should send binding requests to all ESI with matching services. See the "Multiple ESI Application Guidelines" for more details. Hence a device that wishes to receive unsolicited messages from an ESI on the Messaging Cluster, Price Cluster, DRLC Cluster, shall issue a bind request to the ESI for each cluster it is interested in.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

5 A device that sends a binding request is simply announcing itself to an ESI that it desires certain sets of information that the ESI may presently have or may obtain in the future, such as pricing information or DRLC event schedules. The ESIs that receive bind requests are free to refuse them, but if they refuse the binding request, they must choose another method (an address table for instance) to note the device's interest. Once a device has issued the binding request, it does not need to receive a binding response success. If the device receives a NOT_SUPPORTED (or other non-success code) response to a cluster device bind request, it should still send binding requests for any remaining clusters that it has not sent already.

6 After the device has discovered and bound to matching services, it has now established an application layer relationship with all other relevant devices in the HAN (see sub-clause 5.7.2 for details of how to deal with multiple time servers and other duplicated services). That does not mean that the HAN is static and will not acquire new devices, replace devices, or power on devices that were not present during the initial discovery phase. To account for a dynamic HAN, devices shall:

a Repeat the discovery phase on a period of no more than once every three hours and no less than once every 24 hours.

b Repeat the discovery phase after successfully exiting the Rejoin and Recovery Phase (see below).

c Optional - Repeat the discovery phase when a device announce broadcast for a full function device is received. The beginning of the discovery phase should be jittered between 60 and 600 seconds and should be directed only at the device that sent the device announce broadcast.

5.5.5.3 Device Steady State

This is the normal state of the device.

1 A device should make efforts to remain on the correct channel of the PAN and also to keep its network and application keys in sync with the trust center. It is possible that the device has missed a key roll or a channel change due to interference or while it has been powered down or asleep. In order to detect these types of network changes devices shall perform some sort of APS layer message exchange with an ESI on a regular basis. This is to establish that the device can still communicate with the ESI using a current network and APS layer key. This exchange should be performed in accordance with the keep-alive method described in sub-clause 5.4.2.2.3.4. Devices that do not support APS encrypted clusters (Range Extenders for example), do not need to send APS encrypted packets to the Trust Center, but can send network encrypted packets instead.

2 What periodic APS layer message exchange is performed is up to the implementation. Examples would include:

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

a Reading a mandatory Time cluster attribute (such as *CurrentTime*) on the ESI (recommended). This should work for all ESI.

b Reading the current consumption attribute on the Metering cluster (if the ESI supports the Metering server).

c Requesting next pricing info from the ESI (if the ESI supports the Price cluster server).

3 If the device attempts to perform the periodic message exchange and it fails for any reason, the device should note the failure and retry another exchange later. If after no more than twenty-four hours of retries have failed, the device shall go into the Rejoin and Recovery Phase. It is left to the implementation to decide how many retries should occur within the 24 hour period. It is also permissible for the device to enter the Rejoin and Recovery Phase earlier than 24 hours based on number of failed retries or other factors.

4 Sleepy end devices are not required to periodically communicate with an ESI. Instead they should periodically poll their parents and if no parent is found after a suitable period find and rejoin to a new parent. If no parent is found on the original channel, the end device should enter the Rejoin and Recovery phase described below to find a new parent.

5.5.5.4 Rejoin and Recovery State

A device in Rejoin and Recovery Phase is trying to get in sync with its PAN.

1 The device in R&R Phase shall first attempt a ZigBee secure rejoin procedure on its current channel. If the secure rejoin procedure succeeds, the device should revert to its steady state behavior.

2 If the secure rejoin procedure fails, it shall attempt to do a trust center rejoin procedure on its current channel.

3 If the trust center rejoin procedure fails, it may optionally retry steps one and two up to three times.

4 If all attempts to rejoin on the current channel fail, the device shall scan all other channels for its PAN by issuing beacon requests. Note that the PAN ID may have changed and the device shall compare with the extended PAN ID in the beacon and not the short PAN ID.

5 If the device finds an extended PAN ID match in a received beacon, it shall repeat steps one and two on the new channel.

6 If the rejoin (and optional retries) fail on the new channel, the device shall continue scanning all remaining channels for its PAN.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

7 If no correct PANs are discovered on any channel, the device shall return to its original channel to wait for the next R&R attempt.

8 If all rejoin attempts on all channels fail, the device shall return to its original channel to wait for the next R&R attempt. This means that the device is back on the original PAN channel, is still a member of the original PAN, (it has not left the network, and has not discarded any PAN information or security keys), and is simply waiting for the rest of the PAN to appear or to time out and begin another R&R attempt.

9 If while waiting for the next R&R attempt, the device receives an APS encrypted message from an ESI and is encrypted with the device's current network and APS layer key, the device shall leave the R&R phase and proceed to the steady state phase.

10 While in the R&R phase, the device shall retry steps 1-8 periodically, at least once every 24 hours. Sleepy end devices may use a longer period. After four failed rejoin attempts, devices should not try to rejoin any faster than once per hour, with a jitter of +/- 30 minutes.

5.5.5.5 ESI Specific Considerations

1 ESI that are not the PAN coordinator, trust center, or network manager shall perform the steady state phase and rejoin and recovery phase as described above.

2 ESI shall support at a minimum, through bindings or other means, at least five separate devices, with enough resources for each device to bind to all of the relevant clusters that the devices may request bindings to. For example, if the ESI supports five smart energy clusters that devices may send binding requests for, the ESI must support twenty five binding relationships, as well as five sets of device ids and security keys

3 It is strongly recommended that ESI operators remove inactive or deprecated devices from the HAN as well as ESI key and binding tables before adding new devices in order to make room for the new device(s). This use case is an example of a device replacement in the HAN.

4 The Trust Center shall never issue an APS *Remove* command without an explicit request from another device on the network or from the head-end network management system.

5 When a new device is registered with the ESI, the ESI may not have enough resources to support it. If the ESI is low on resources, it should notify the installer or ESI administrator (this could be via user interface, or backhaul/ backchannel communication for example.) The ESI shall not automatically remove other devices in order to free up resources for the new device without explicit approval from the installer or ESI administrator.

6 If a device joins the PAN, but does not successfully perform Key Establishment, a trust center may remove the device. This shall only be done after more than 1 hour has elapsed since the device's initial join. This shall be sent directly to the router, or to the parent of an end device. A

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

child that receives a NWK leave from its parent when it does not have an authorized link key (i.e. not performed key establishment successfully) shall not ignore the leave.

7 It is permissible and encouraged that an ESI perform its own service discovery procedure after power up and on a periodic basis. The ESI may independently create its own bindings to devices with matching services. This may in some cases establish application layer relationships faster than waiting for devices to request bindings by themselves. As specified in sub-clause 5.4.5.1, an ESI that does not support bindings shall perform its own service discovery.

5.6 Federated Trust Center Application Guidelines

Note: *The Federated Trust Center feature in this revision of this specification is provisionary and is not certifiable. This feature set may change before reaching certifiable status in a future revision of this specification.*

When considering Smart Energy systems to support multiple dwelling units (MDUs) such as apartment blocks, one solution is to use a single ZigBee mesh network (NAN) to support the entire building. In this scenario, there will be a single Trust Center (ESI), also known as a *Federated Trust Center*, that will control this entire network.

The devices associated with a particular apartment within the MDU will form a ‘virtual HAN’, and will consist of devices within the apartment together with the associated meter(s) which may be located in a central meter room or elsewhere within the building. Thus the overall network for the entire building will be made up of a number of groups of devices.

The Federated Trust Center will need to be pre-loaded with information indicating the devices that will form the ‘virtual HAN’ for each apartment. Additional functionality will be required so that a device joining this NAN can be made aware of the other devices that will constitute the virtual network for the relevant apartment; the device will then limit its subsequent service discovery to that list of devices.

Once a device has joined the NAN, and has successfully completed key establishment with the Trust Center, it shall request Pairing Information from the Trust Center. Once Pairing Information is returned, the device shall limit service discovery to those devices identified in the response. It should be noted that, in this scenario, service discovery will be performed using unicast commands rather than broadcast.

Devices may join and leave the ‘virtual HAN’ from time to time. To allow for this possible reorganization, each device within the NAN shall periodically ask the Federated Trust Center

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

whether new Pairing Information is available. The typical period is once every 24 hours. A *Version* field within the relevant commands allows updated information to be identified.

For further information, refer to the MDU Pairing cluster definition in Annex D.13.

5.7 Multiple ESI Application Guidelines

5.7.1 Overview

The ZigBee Smart Energy Profile allows for the use of multiple ESIs in a HAN. This feature is desirable from a reliability perspective, plus opens opportunities for vendors to innovate and provide additional services and functionality. Multiple ESIs does not mean multiple Trust Centers, only a single Trust Center is supported in a HAN.

Clients may assume that all SE messages/directives (Demand Response events, price publishing, messaging) are created by the same entity, e.g., utility or energy management entity, or set of coordinated entities. These messages can be sent to devices via one or more transport mechanisms (in the HAN, this means the same message may be sent from multiple ESIs). A message with a specific ID typically will be unique within the system, even though a device may receive this message more than once. However, in a HAN with multiple, uncoordinated commodity service providers (e.g., gas vs. water, household electricity vs. PEV electricity), there is a possibility that different, unique events will have conflicting event IDs. Since it is expected that ID conflicts for events occurring at similar points of time will be rare, clients may ignore the issue and always assume that conflicting event IDs are duplicates. More complex clients may choose to better track events by service provider, commodity type, etc.

Note: *The functionality in this section does not describe the operation of multiple ESIs within a MDU/NAN. Refer to section 5.6 for details of Federated Trust Center functionality.*

5.7.2 Device Behavior

5.7.2.1 Service Discovery in Multi ESI Environments

A device should make itself aware of any and all ESIs in a SE HAN using service discovery. It shall perform this service discovery upon joining a network, power up (and network rejoin), and periodically. The typical period is once every 24 hours. A device that discovers an ESI with matching services shall create bindings on the ESI so that the ESI will register the device and send it appropriate SE commands. Devices which do not bind in a multiple ESI network are expected to poll the ESIs. ESIs that are not rediscovered over the period of multiple discovery cycles may be forgotten by the device.

A device that discovers more than one ESI should determine a single ESI as an authoritative time source. To do so, it should use the Time cluster Master, Synchronized, and Superseding bits.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

A Time server with the Superseding bit set will always take precedence over a Time server without that bit set, including ones that have the Master bit set. A new ESI going into a faulty installation can set the Superseding bit and take over the network's Time synchronization. However, it is not required for SE 1.x to have this bit set if the new ESI does not want to forcefully take over the Time server role. This bit is set independently of the other three *TimeStatus* bits (Master, Synchronized, MasterZoneDst).

5.7.2.2 Determining the Most Authoritative Time Source

Devices shall synchronize to a Time server with the highest rank according to the following rules, listed in order of precedence:

1. A server with the Superseding and Master bits set shall be chosen over a server with only the Master bit set.
2. A server with the Master bit set shall be chosen over a server without the bit set.
3. The server with the lower short address shall be chosen (note that this means a coordinator with the Superseding and Master bit set will always be chosen as the network time server).
4. A Time server with neither the Master nor Synchronized bits set should not be chosen as the network time server.

5.7.2.3 Periodic Time Source Checking During Normal Operation

During normal operation (the most authoritative time source is found and it has valid time), clients periodically repeat the time source scan to pick up new, more authoritative time sources, as per the following rules:

1. Non-sleepy clients shall locate the most authoritative time source at least once every 24 hours.
2. Sleepy devices should locate the most authoritative time source at least once every 24 hours.
3. Clients shall scan for time sources after rebooting, and after joining or rejoining the network.

5.7.2.4 Invalid Time and Interim Time Sources

Although the rules above are used to find the most authoritative time source for the network, there are conditions where what would normally be the most authoritative source is temporarily unable to provide valid time. In this situation, regardless of whether it is encountered as part of the original time discovery or the periodic rediscovery, devices obey the following rules:

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

1. If a server is temporarily unable to provide valid UTC, it shall report all time attributes (e.g., UTC, local time) as 0xFFFFFFFF. It should leave the Superseding, Master, and Synchronized bits set as if it did have valid time.

2. If the most authoritative time source for the network has invalid time, clients should temporarily use the Time server of next highest rank, but shall periodically look for the more authoritative server(s) to obtain valid time. Non- sleepy devices shall check at least once every 15 minutes. Sleepy devices may check as often as their power budget will allow.

3. When a more authoritative time source with valid time is found, clients shall immediately switch to using that source's time basis.

5.7.2.5 Handling SE Commands from Multiple ESIs

When a device creates bindings on multiple ESIs, it may receive SE commands from those ESIs. Simple device logic such as assuming all commands came from the PAN coordinator is not appropriate. The following rules describe the desirable device behavior.

1. When a device receives an event (Demand Response, Price, Messaging) any time reference in the message should be viewed in context with the time reference of the most authoritative ESI time server.

2. When a device receives duplicate events (same event ID) from multiple ESIs, it shall send an event response to each ESI. Future duplicate events from the same ESI(s) shall be either “ignored” by sending no response at all or with a default response containing a success status code.

3. Conflicting events with the same event ID from different ESIs will be resolved by the device in the same manner as if they came from a single ESI.

4. When a device has an asynchronous follow up event response it should send the response to the ESIs that created the condition. If the event was received from more than one ESI, the device shall send the asynchronous event response to all ESIs from which it received the event.

5.7.2.6 Handling Multiple Uncoordinated Back-end Systems

When multiple, uncoordinated service providers deploy ESIs in a HAN, it is possible that different back-ends/ESIs will have a different notion of time. However, only one ESI will be the authoritative time source for the HAN. In this scenario, an ESI may require a mechanism to ensure that its events are executed on its time basis, even if it is not the authoritative time source. ESIs that require this behavior may make use of the following application guidelines:

1. An ESI may implement the Time cluster client and determine the most authoritative time source for the HAN, using the rules defined above.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

2651 2652 2653 **2** If the ESI is not the authoritative time source, it may synchronize its clock, or apply differentials to the start time of its events, to ensure that clients execute those events on the intended schedule.

2654 2655 2656 **3** If the more authoritative time server disappears and is not seen for 24 hours, clients may assume that the server has left the network, and resume normal operation using the most authoritative time server (with valid time) that remains.

2657 ## 5.8 Other Smart Energy Profile Requirements and Best Practices

2658 ### 5.8.1 Preferred Channel Usage

2659 2660 2661 2662 When forming a new network, or scanning to join a network, Smart Energy devices should do channel scans using the following preferred channels before scanning the rest of the channels in order to avoid the most commonly used WiFi channels. This is to improve the user experience during installation (quicker joining) and possibly improve bandwidth (on average).

2663 **Preferred 2.4 GHz Channels** - 11, 14, 15, 19, 20, 24, 25

2664 **Preferred 900MHz Channels** – Use all available for ZigBee.

2665 ### 5.8.2 Broadcast Policy

2666 2667 2668 Broadcasts are strongly discouraged for Smart Energy devices. Devices are limited to a maximum broadcast frequency of one broadcast per second and strongly encouraged to exercise broadcasts much less frequently.

2669 ### 5.8.3 Frequency Agility

2670 2671 2672 2673 2674 Frequency Agility would only be officially exercised in a network by a system controller, or higher functioning device (ESI, aggregator, installation tool, etc…). Devices may support frequency agility hooks to be commanded to "go to channel X". Devices that do not support frequency agility may implement either the NWK rejoin or orphan join feature to find a network that has changed channels.

2675 ### 5.8.4 Key Updates

2676 2677 2678 2679 Energy devices are only required to support ZigBee "residential mode" security or ZigBee PRO "standard mode" with the required use of link keys. All link key updates shall use the Key Establishment Cluster. Sleeping devices that miss key updates can request a new key using the existing link key so there is no problem with sleeping devices missing key updates.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

5.9 Coexistence and Interoperability with HA Devices

It is desirable to allow interoperability of HA and Smart Energy devices where practical. However, it is undesirable to publicly share keys during the joining process or share private information over a less secure network. HA devices that only provide functionality for receiving network keys in the clear during a join process cannot be used in a Smart Energy network. HA devices may also be extended with Smart Energy clusters providing they support the use of Link Keys and the Smart Energy security models. If so, they can be certified as HA and Smart Energy capable allowing those devices to operate either in an HA network or a Smart Energy network.

5.10 Device Descriptions

Device descriptions specified in this profile are summarized in Table 5-14 along with their respective Device IDs. The devices are organized according to the end application areas they address. A product that conforms to this specification shall implement at least one of these device descriptions and shall also include the device descriptions corresponding to all applications implemented on the product where a standard device description is specified in this profile. For example, if a product implements both a thermostat and an In-Home Display, then the thermostat and In-Home Display device descriptions must both be supported.

This list will be added to in future versions of the profile as new clusters are developed to meet the needs of manufacturers. The reserved values shall not be used until the profile defines them. Manufacturer-specific device descriptions shall reside on a separate endpoint and use a private profile ID.

Table 5-14 – Devices Specified in the Smart Energy Profile

	Device	Device ID
Generic	Range Extender	0x0008
Smart Energy	Energy Service Interface	0x0500
	Metering Device	0x0501
	In-Home Display	0x0502
	Programmable Communicating Thermostat	0x0503
	Load Control Device	0x0504
	Smart Appliance	0x0505

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

2702

	Prepayment Terminal	0x0506
	Physical Device	0x0507
	Remote Communications Device	0x0508
	Reserved	0x0509 – 0x5FF

2703
2704

2705 5.11 ZigBee Cluster Library (ZCL)

2706 This profile utilizes some of the clusters specified in the ZigBee Cluster Library. The
2707 implementation details for each cluster are given in the ZCL specifications. Further
2708 specification and clarification is given in this profile where necessary.

2709 The ZCL provides a mechanism for clusters to report changes to the value of various
2710 attributes. It also provides commands to configure the reporting parameters. Products shall
2711 support the attribute reporting mechanism for supported attributes as specified in the ZCL. The
2712 minimum reporting interval specified in the ZCL [B1] shall be set to a value greater than or
2713 equal to 0x0001. The maximum reporting interval should be set to 0x0000 by default, and if it is
2714 set to a non-zero value it shall be set to a value greater than or equal to 0x003C and greater than
2715 the value of the minimum reporting interval. These settings will restrict the attributes from
2716 being reported more often than once every second if the attribute is changing quickly and at least
2717 once every minute if the attribute does not change for a long time. It is recommended that the
2718 minimum reporting interval be set to one minute and the maximum reporting interval be set to a
2719 much greater value to avoid unnecessary traffic.

2720 Devices shall use the ZCL default response error handing. Typical examples of this are:

2721 • When receiving commands that don't have data collected such as Get Scheduled Events, Get
2722 Current Price, Get Scheduled Prices, Get Block Period(s), and Get Last Message, devices
2723 shall respond using the ZCL default response with a status code of NOT_FOUND.

2724 • When receiving requests for unsupported commands, devices shall respond using the ZCL
2725 default response with a status code of UNSUP_CLUSTER_COMMAND.

2726 • When receiving malformed commands, devices shall respond using the ZCL default
2727 response with a status code of MALFORMED_COMMAND.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

- When receiving requests for accessing unsupported attributes, devices shall respond using the ZCL default response with a status code of UNSUPPORTED_ATTRIBUTE.

- When receiving requests for accessing attributes where there is an additional access control mechanism and access is denied, devices shall respond using the ZCL default response with a status code of NOT_AUTHORIZED. Any additional access control mechanism is optional and the specification of the additional access control mechanism is out of scope of this specification.

Please refer to [B1] for additional status codes support in the ZCL default response.

5.12 Cluster List and IDs

The clusters used in this profile are listed in Table 5.15. The clusters are listed according to the functional domain they belong to in the ZCL and indicate the additional new Smart Energy clusters. The existing corresponding ZCL General cluster identifiers can be found in the ZCL [B1].

The functionality made available by all supported clusters shall be that given in their ZCL specifications except where a device description in this profile includes further specification, clarification or restriction as needed for a particular device.

Most clusters include optional attributes. The application designer must be aware that optional attributes might not be implemented on a particular device. All Smart Energy devices must discover and deal with unsupported attributes on other devices.

It is expected that clusters will continue to be developed in the ZCL that will be useful in this profile. In many cases, new clusters will be organized into new device descriptions that are separate from those currently defined. There may also be situations where it makes sense to add clusters as optional elements of existing device descriptions.

Manufacturer-specific clusters may be added to any device description in this profile as long as they follow the specifications given in the ZCL [B1].

Table 5-15 – Clusters Used in the Smart Energy Profile

Functional Domain	Cluster Name	Cluster ID
General	Basic	0x0000
General	Identify	0x0003
General	Alarms	0x0009
General	Time	0x000A
General	Commissioning	0x0015

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

General	Power Configuration	0x0001
General	Key Establishment	0x0800
Smart Energy	Price	0x0700
Smart Energy	Demand Response and Load Control	0x0701
Smart Energy	Metering	0x0702
Smart Energy	Messaging	0x0703
Smart Energy	Smart Energy Tunneling (Complex Metering)	0x0704
Smart Energy	Prepayment	0x0705
Smart Energy	Energy Management	0x0706
Smart Energy	Calendar	0x0707
Smart Energy	Device Management	0x0708
Smart Energy	Events	0x0709
Smart Energy	MDU Pairing	0x070A

5.12.1 ZCL General Clusters

Except for the Key Establishment Cluster, which is covered in Annex C, please refer to the ZCL Cluster Specification [B1] for the General Cluster descriptions.

5.12.1.1 ZCL Time Cluster and Time Synchronization

The Smart Energy profile requires time synchronization between devices to properly support the coordination of Demand Response/Load Control events, Price changes, and the collection of metered data. In order to simplify the understanding of time, the Smart Energy profile will leverage UTC as the common time base. To this end a new ZCL attribute data type, UTCTime is included and its definition can be found in Annex A.

It is desired for the processes for synchronizing time to be as network friendly as possible to eliminate excessive traffic. To support this, time accuracy on Client devices shall be within +/- 1 minute of the server device (ESI) per 24 hour period. The Client devices shall design a clock accuracy that never requires more than one time synchronization event per 24 hour period. The exception to this is when devices need to rejoin or re-register on the network. Again, the desire is to keep time synchronization traffic to a minimum.

Further, implementers must be aware that network communication delays will cause minor differences in time between devices. The Smart Energy profile expectations are that this will be a minor issue given the use cases it's fulfilling. It will not nor does it recommend

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

implementers develop an NTP or equivalent scheme to compensate for network delays. These methods are viewed as having the potential to cause excessive network communications.

5.12.1.2 Transaction Sequence Numbers

The normal usage of Transaction Sequence Numbers is defined in [B1] 2.3.1.3. The following clarifications shall be noted with respect to the usage of Transaction Sequence Numbers within the Smart Energy Standard:

1 Where a 'Publish' command is sent in response to the receipt of an associated 'Get' command, the 'Publish' command shall use the Transaction Sequence Number received in the 'Get' command

2 Where there is more than one response to a request, the Transaction Sequence Number from the request shall be used in all responses to that specific request

3 Where a 'Publish' command is sent unsolicited, the server is free to choose any value of Transaction Sequence Number

A Transaction Sequence Number is not to be used to reject a command, only to correlate messages.

5.13 Coexistence with devices using other Profiles

Note: *The Profile Coexistence feature in this revision of this specification is provisionary and is not certifiable. This feature set may change before reaching certifiable status in a future revision of this specification.*

Devices that do not implement Smart Energy 1.x clusters may be allowed on to the ZigBee network at the discretion of the network owner. Additional router devices have the benefit of growing the range and reliability of the ZigBee network, and offering additional services not offered by the Smart Energy profile. This also benefits the user as it means fewer ZigBee networks to manage.

This section describes only how Smart Energy devices and devices from other profiles will interact in a general sense to join the network and communicate. It does not describe how specific clusters from one profile may be used to communicate with clusters on a different profile.

All devices in the Smart Energy protocol are required to adhere to these requirements regardless of whether the network owner chooses to allow devices from other profiles on their network.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

5.13.1 Requirements for the Smart Energy Trust Center

1 The Trust Center shall only allow devices onto the Smart Energy network that utilize installation code based link keys. This is the normal behavior per the Smart Energy profile.

2 If a Trust Center does not want to allow non-Smart Energy devices onto the network, it is recommended that the TC should check whether a joining device supports the Key Establishment cluster and, if not, request the joining device to leave the network.

3 It is recommended that Smart Energy devices that fail CBKE should remove themselves from the network.

4 Trust Centers that support both Smart Energy and non-Smart Energy devices should not remove non-Smart Energy devices that do not attempt CBKE.

5 If the Trust Center sends NWK key updates via unicast, then it shall still include devices from other profiles that have not performed CBKE.

6 Per normal requirements, the Trust Center shall not grant partner link keys to devices that have not performed CBKE.

7 The Trust Center shall use a device's install code key to transport the NWK key if a Trust Center rejoin is performed by a device from another profile.

5.13.2 Requirements for Smart Energy Devices

1 Smart Energy endpoints that receive Smart Energy cluster messages using a profile other than the Smart Energy profile shall drop the messages.

2 Smart Energy endpoints must continue to follow the cluster security requirements specified in Table 5-13 and drop Smart Energy messages that do not have APS security when it is required (refer to section 5.4.6 for further details).

3 Smart Energy endpoints that also implement clusters from another profile shall not use their install code key for any encryption or decryption once they establish a CBKE link key.

4 Smart Energy endpoints shall not refer to a Smart Energy-related cluster on a non-Smart-Energy endpoint. Note that this does not preclude a non-Smart-Energy version of a cluster existing on a HAN as well as a Smart Energy-related version.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

5.13.3 Requirements for Devices from other Profiles

5.13.3.1 Joining

1 Devices from other profiles must implement install code based link keys. This includes proper documentation of the install code via packaging and instructions provided to the installer. Refer to section 5.4.8 for further details.

2 The Smart Energy network must be informed of the device that is to be joined. This is done through an out of band means which could include a web login, phone call to a service center, or handheld tool. Using this methodology the existing network is made aware of the device ID and Install Code/Link Key for the device.

3 The Smart Energy network is put into permit joining ON for a period of time.

4 The installer/homeowner is prompted to press a button or complete a menu sequence that tells the device to attempt to join a network.

5 The device joins the network.

6 An indicator is provided for the installer/homeowner indicating the device has joined a network.

7 The device can now operate normally on the network according to the rules of the relevant profile.

8 Devices from other profiles shall have no expectations they may use their link key with the Trust Center other than to encrypt/decrypt APS command frames. Their link key will be treated like an unauthorized link key. The Trust Center shall not accept APS data frames, for Smart Energy clusters requiring APS encryption, where such data frames are encrypted using an unauthorized link key.

9 Currently there is no provision in this specification for a device from another profile to obtain an authorized link key.

5.13.3.2 Rejoining

Devices from other profiles may perform a secure or trust center rejoin to get back on the network. A trust center rejoin shall use the device's install code key.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

6 Device Specifications

6.1 Common Clusters

Support for certain clusters is required on all products supporting this profile. At least one instance of the clusters shown in Table 6-1 shall exist on a product supporting this profile. Individual device descriptions may place further restrictions on support of the optional clusters shown here. ZCL clusters not listed may be implemented on the Smart Energy endpoint. Manufacturers may extend the SE profile as specified in the ZCL specification [B1].

Table 6-1 – Clusters Common to All Devices

Server Side	Client Side
Mandatory	
Basic	*None*
Key Establishment	Key Establishment
Optional	
Power Configuration	None
Commissioning	Commissioning
Identify	*None*
Events	
OTA Upgrade	OTA Upgrade

6.1.1 Optional Support for Clusters with Reporting Capability

Some clusters support the ability to report changes to the value of particular attributes. These reports are typically received by the client side of the cluster. All devices in this profile may support any cluster that receives attribute reports.

6.1.2 Manufacturer-Specific Clusters

The ZCL provides a range of cluster IDs that are reserved for manufacturer- specific clusters. Manufacturer-specific clusters that conform to the requirements given in the ZCL may be added to any device description specified in this profile.

6.1.3 Cluster Usage Restrictions

None.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

6.1.4 Identify Cluster Best Practices

To help aid in locating devices, it's strongly recommended that all devices utilize the Identify Cluster and a visual or audible indicator. In situations in which a device can't supply a visual or audible indicator, the device should include a visible label with the appropriate information to help identify the device.

6.1.5 Inter-PAN Communication

Inter-PAN access to Smart Energy devices shall be limited to specific clusters and commands. Please refer to Annex B for further details.

6.2 Feature and Function Description

Each device must support a certain set of features and functions. Table 6-2 below is used to specify the mandatory and optional features and functions for Smart Energy devices. This chapter contains a description of what must be supported if the feature or function is supported by the device. The mandatory or optional configuration for each device is described in the upcoming chapters:

Table 6-2 – Common Features and Functions Configuration for a Smart Energy Device

Device Type/ Feature or function	Join (end devices and routers only)	Form Network (coordinator only)	Restore to Factory Fresh Settings	Pair Devices – (End Device Bind Request)	Bind Manager – (End Device Bind Response - Coordinator only)	Enable Identify Mode	Allow Smart Energy devices to join the Network (routers and coordinators only)
Mandatory/ Optional	M	M	M	O	M	O	M
Device Type/ Feature or function	Service discovery (Match Descriptor Request)	ZDP Bind Response	ZDP Unbind Response	End Device Annce/ device annce	Service Discovery response (Match Descriptor Response)	High Security Supported (ZigBee PRO only)	Enhanced Inter-PAN Communication
Mandatory/ Optional	O	M	M	M	M	N/A	O

Join (End Devices and Routers):

As described in sub-clauses 5.4 and 5.5.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Form Network (Coordinator):
As described in sub-clauses 5.4 and 5.5.

Allow Others to Join Network (Router and Coordinator Only):
As described in sub-clauses 5.4 and 5.5.

Restore to Factory Fresh Settings:
The Device shall provide a way to restore Factory Settings.

Pair Devices (End Device Bind Request):
Whenever possible, the device should provide a way for the user to issue an End Device Bind Request.

Bind Manager (End Device Bind Response – Coordinator only):
The coordinator device shall be capable of issuing an End Device Bind Response.

Enable Identify Mode:
Whenever possible, the device should provide a way for the user to enable Identify for 60 seconds.

Service Discovery (Match Descriptor Request):
Whenever possible, the device should provide a way for device to send a match descriptor request, receive match descriptor responses and utilize them for commissioning the device.

ZDP Bind Response:
The device shall be able to receive a ZDP Bind Request and respond correctly with a ZDP Bind Response.

ZDP Unbind Response:
The device shall be able to receive a ZDP Unbind Request and respond correctly with a ZDP Unbind Response.

End Device Annce/Device Annce:
The device shall Send End Device Annce / Send Device upon joining and re- joining a network.

Service Discovery Response:
The Device shall be able to receive a Match descriptor request, and respond with a match descriptor response correctly.

Allow Smart Energy Devices to Join the Network:
The Device shall allow other Smart Energy devices to join the network.

High Security Supported: No

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Enhanced Inter-PAN Communication:

The device may support Enhanced Inter-PAN Communications as described in Annex B

6.3 Smart Energy Devices

A physical device may support one or more logical Smart Energy devices. The supported clusters of a logical Smart Energy device shall reside on a single endpoint, with the exception of the common clusters listed in Table 6-1, which may reside on a separate endpoint using the Physical Device identifier. Each logical Smart Energy device on a single physical device shall reside on its own separate endpoint.

SE devices shall use the device and service discovery mechanisms specified in the ZigBee specification [B3] to find the services required. Devices shall support discovery of single and multiple endpoints on a single physical device. In the case where multiple devices of the same type are discovered, SE cluster attributes should be read to determine the type of service provided. For example, if multiple Metering devices are discovered the *MeteringDeviceType* attribute provides a label for identifying the type of metering device present. Similarly, if multiple ESIs are found the *CommodityType* attribute shall be read to determine the fuel type of that ESI.

6.3.1 Energy Service Interface

The Energy Service Interface connects the energy supply company communication network to the metering and energy management devices within the home. It routes messages to and from the relevant end points. It may be installed within a meter, thermostat, or In-Home Display, or may be a standalone device, and it will contain another non-ZigBee communication module (e.g. power-line carrier, RF, GPRS, broadband Internet connection).

6.3.1.1 Supported Clusters

In addition to those specified in Table 6-1, the Energy Service Interface device shall support the clusters listed in Table 6-3. If a SE cluster is not listed as mandatory or optional in the following table or in the common table, then that cluster shall be prohibited on an ESI device endpoint.

Table 6-3 – Clusters Supported by the Energy Service Interface

Server Side	Client Side
Mandatory	
Messaging	
Price	
Demand Response/Load Control	
Time	

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Optional	
	Price
Calendar	
Metering	Metering
Prepayment	Prepayment
	Time
Device Management	Device Management
Alarms	
	Events
MDU Pairing	MDU Pairing
	Energy Management
Tunneling	Tunneling

6.3.1.2 Supported Features and Functions

The Energy Service Interface device shall have the features and functions listed in Table 6-2.

6.3.2 Metering Device

The Metering end device is a meter (electricity, gas, water, heat, etc.) that is fitted with a ZigBee device. Depending on what is being metered, the device may be capable of immediate (requested) reads or it will autonomously send readings periodically. A Metering end device may also be capable of communicating certain status indicators (e.g. battery low, tamper detected).

6.3.2.1 Supported Clusters

In addition to those specified in Table 6-1, the Metering Device shall support the clusters listed in Table 6-4. If a SE cluster is not listed as mandatory or optional in the following table or in the common table, then that cluster shall be prohibited on a Metering device endpoint.

Table 6-4 – Clusters Supported by the Metering Device

Server Side	Client Side
Mandatory	
Metering	
Optional	
	Time

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Prepayment	
	Price
	Calendar
	Messaging
	Device Management
	MDU Pairing
Alarms	
Tunneling	Tunneling

2977

2978 #### 6.3.2.2 Supported Features and Functions

2979 The Metering Device shall have the features and functions listed in Table 6-2.

2980

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

6.3.3 In-Home Display Device

The In-Home Display device will relay energy consumption data to the user by way of a graphical or text display. The display may or may not be an interactive device. At a minimum at least one of the following should be displayed: current energy usage, a history over selectable periods, pricing information, or text messages. As an interactive device, it can be used for returning simple messages for interpretation by the recipient (e.g. “Button A was pressed”).

The display may also show critical pricing information to advise the customer when peaks are due to occur so that they can take appropriate action.

6.3.3.1 Supported Clusters

In addition to those specified in Table 6-1, the In-Home Display device shall support the clusters listed in Table 6-5. If a SE cluster is not listed as mandatory or optional in the following table or in the common table, then that cluster shall be prohibited on an In-Home Display device endpoint.

Table 6-5 – Clusters Supported by the In-Home Display Device

Server Side	Client Side
Mandatory	
Optional	
	Demand Response and Load Control
	Time
	Prepayment
	Price
	Calendar
	Metering
	Messaging
	Device Management
	MDU Pairing
	Energy Management
Alarms	
	Events
Tunneling	Tunneling

An In-Home Display shall implement at least one of the optional client clusters listed.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

6.3.3.2 Supported Features and Functions

The In-Home Display device shall have the features and functions listed in Table 6-2.

6.3.4 Programmable Communicating Thermostat (PCT) Device

The PCT device shall provide the capability to control the premises heating and cooling systems.

6.3.4.1 Supported Clusters

In addition to those specified in Table 6-1, the PCT device shall support the clusters listed in Table 6-6. If a SE cluster is not listed as mandatory or optional in the following table or in the common table, then that cluster shall be prohibited on a PCT device endpoint.

Table 6-6 – Clusters Supported by the PCT

Server Side	Client Side
Mandatory	
	Demand Response and Load Control
	Time
Optional	
	Prepayment
	Price
	Calendar
	Metering
	Messaging
	Device Management
	MDU Pairing
Energy Management	
Alarms	
Tunneling	Tunneling

6.3.4.2 Supported Features and Functions

The PCT device shall have the features and functions listed in Table 6-2.

6.3.5 Load Control Device

The Load Control device is capable of receiving Demand Response and Load Control events to manage consumption on a range of devices. Example devices are water heaters, exterior lighting, and pool pumps.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

6.3.5.1 Supported Clusters

In addition to those specified in Table 6-1, the Load Control device shall support the clusters listed in Table 6-7.

Table 6-7 – Clusters Supported by the Load Control Device

Server Side	Client Side
Mandatory	
	Demand Response and Load Control
	Time
Optional	
	Price
	Calendar
	Device Management
	MDU Pairing
Energy Management	
Alarms	
Tunneling	Tunneling

6.3.5.2 Supported Features and Functions

The Load Control Device shall support the features and functions listed in Table 6-2.

6.3.6 Range Extender Device

The Range Extender is a simple device that acts as a router for other devices. The Range Extender device shall not be a ZigBee end device. A product that implements the Range Extender device shall not implement any other devices defined in this profile. This device shall only be used if the product is not intended to have any other application, or if a private application is implemented that has not been addressed by this profile.

6.3.6.1 Supported Clusters

The Range Extender device shall support the mandatory common clusters listed in Table 6-1.

6.3.6.2 Supported Features and Functions

The Range Extender device shall have the features and functions listed in Table 6-2.

6.3.7 Smart Appliance Device

Smart Appliance devices on the ZigBee network can participate in energy management activities. Examples of these are when Utilities initiate a demand response or pricing event, or

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

the appliance actively informs customers via in- home displays of when or how energy is being used. In the latter case, scenarios include:

- Washer switching to cold water during periods of higher energy costs.
- Washer/Dryer/Oven/Hot Water Heater reporting cycle status.
- Over temperature conditions in Freezers and Refrigerators.

6.3.7.1 Supported Clusters

In addition to those specified in Table 6-1 the Smart Appliance device shall support the clusters listed in Table 6-8. If a SE cluster is not listed as mandatory or optional in the following table or in the common table, then that cluster shall be prohibited on a Smart Appliance device endpoint.

Table 6-8 – Clusters Supported by the Smart Appliance Device

Server Side	Client Side
Mandatory	
	Price
	Time
Optional	
	Demand Response and Load Control
	Messaging
	Calendar
	Device Management
	MDU Pairing
Energy Management	
Alarms	
Tunneling	Tunneling

6.3.7.2 Supported Features and Functions

The Smart Appliance device shall have the features and functions listed in Table 6-2.

6.3.8 Prepayment Terminal Device

The Prepayment Terminal device will allow utility customers or other users (e.g. sub-metered tenants) to pay for consumption in discrete increments rather than establishing a traditional

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

billing agreement. The Prepayment Terminal device will accept payment (e.g. credit card, code entry), display remaining balances, and alert the user of a balance approaching zero, and may perform some or all of the other functions described in sub-clause 6.3.3 “In-Home Display Device”.

6.3.8.1 Supported Clusters

In addition to those specified in Table 6-1, the Prepayment Terminal device shall support the clusters listed in Table 6-9. If a SE cluster is not listed as mandatory or optional in the following table or in the common table, then that cluster shall be prohibited on a Prepayment Terminal device endpoint.

Table 6-9 – Clusters Supported by the Prepayment Terminal Device

Server Side	Client Side
Mandatory	
	Price
	Time
Prepayment	Prepayment
Optional	
	Demand Response and Load Control
	Calendar
	Metering
	Messaging
	Device Management
	MDU Pairing
	Energy Management
Alarms	
Tunneling	Tunneling

6.3.8.2 Supported Features and Functions

The Prepayment Terminal device shall have the features and functions listed in Table 6-2.

6.3.9 Physical Device

The Physical Device type will identify a supplemental (or sole) endpoint on which the clusters related to a physical product may reside. The endpoint shall not contain any cluster related to any individual logical SE device on the physical product. A product is allowed to have a Physical Device as its sole SE endpoint. A Physical Device must be capable of providing other SE device endpoints to be a certified SE product.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

3073 #### 6.3.9.1 Supported Clusters

3074 The Physical Device may only support the common clusters listed in Table 6-1.

3075 #### 6.3.9.2 Supported Features and Functions

3076 The Physical Device shall have the features and functions listed in Table 6-2.

3077

3078 ### 6.3.10 Remote Communications Device

3079 The Remote Communications Device may be used to provide a WAN communication path
3080 through the ZigBee HAN where an ESI does not have a direct WAN connection. The Remote
3081 Communications Device endpoint may reside on a physical device with a direct connection to the
3082 WAN, allowing it to tunnel WAN protocol messages to an ESI which does not have a direct
3083 WAN connection.

3084 #### 6.3.10.1 Supported Clusters

3085 In addition to those specified in Table 6-1, the Remote Communications Device shall support
3086 the mandatory clusters listed in Table 6-10 and may support the optional clusters. If a SE cluster
3087 is not listed as mandatory or optional in the following table or in the common table, then that
3088 cluster shall be prohibited on a Remote Communications Device endpoint.

3089 **Table 6-10 – Clusters Supported by the Remote Communications Device**

Server Side	Client Side
Mandatory	
Optional	
Tunneling	Tunneling
Time	

3090
3091 A Remote Communications Device shall implement at least one of either the Tunneling cluster
3092 Server or Client.

3093 Provision of the optional Time server allows support of an accurate commodity-agnostic time
3094 source, if such a source were available via the WAN connection.

3095 #### 6.3.10.2 Supported Features and Functions

3096 The Remote Communications device shall have the features and functions listed in Table 6-2.

3097

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

3098

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Annex A CANDIDATE ZCL MATERIAL FOR USE WITH THIS PROFILE

The candidate material in this annex, when approved, will be merged into the Foundation document of the ZigBee Cluster Library (ZCL) by the Cluster Library Development Board.

A.1 New Status Enumeration

A new ZCL status enumeration is required for use with Smart Energy devices, as described in Table A-1.

Table A-1– Additional Status Enumeration

Enumerated Status	Value	Description
NOTIFICATION_PENDING	0x9A	The command has been received and is being processed.

A.2 New Attribute Reporting Status Indication

A new *Attribute Reporting Status* attribute is to be added to all Smart Energy clusters. Consideration will be given to extending this to all ZCL clusters. Details for this attribute are shown in Table A-2:

Table A-2– *Attribute Reporting Status* Attribute

Identifier	Name	Type	Range	Access	Default	Man. /Opt.
0xFFFE	*Attribute Reporting Status*	8-bit Enumeration	0x00 to 0xFF	Read Only	-	O

A.2.1 *Attribute Reporting Status* Attribute

When utilizing the *Report Attributes* command, this 8-bit enumeration should be included as the last attribute in order to indicate that all required attributes have been reported. The enumerated values for this attribute are outlined in Table A-3:

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

3118 **Table A-3– Attribute Reporting Status Enumerations**

Enumerated Value	Status
0x00	Pending
0x01	Attribute Reporting Complete
0x02 to 0xFF	Reserved for future use

3119

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Annex B ENHANCED INTER-PAN TRANSMISSION MECHANISM

NOTE: This annex previously described the requirements for the 'Anonymous Inter-PAN' mechanism. It has been removed and replaced with a new annex defining the 'Enhanced' Inter-PAN mechanism which is to be used in conjunction with devices such as Hand Held Terminals (HHTs).

The 'Enhanced' Inter-PAN mechanism in this revision of this specification is provisionary and is not certifiable. This mechanism may change before reaching certifiable status in a future revision of this specification.

B.1 Scope and Purpose

This annex defines a mechanism whereby ZigBee devices can perform limited exchanges of information with devices in their local neighborhood without having to form or join the same ZigBee network. The Enhanced Inter-PAN mechanism is intended to be used in conjunction with devices such as Hand Held Terminals (HHTs).

The intended destination for the mechanism described here is not the ZigBee specification [B7], but the relevant application profile documents for applications that make use of the feature – in particular, the Smart Energy Standard.

B.2 General Description

B.2.1 What Enhanced Inter-PAN Transmission Does

A schematic view of the how inter-PAN transmission in a ZigBee context works is shown in Figure B-1.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

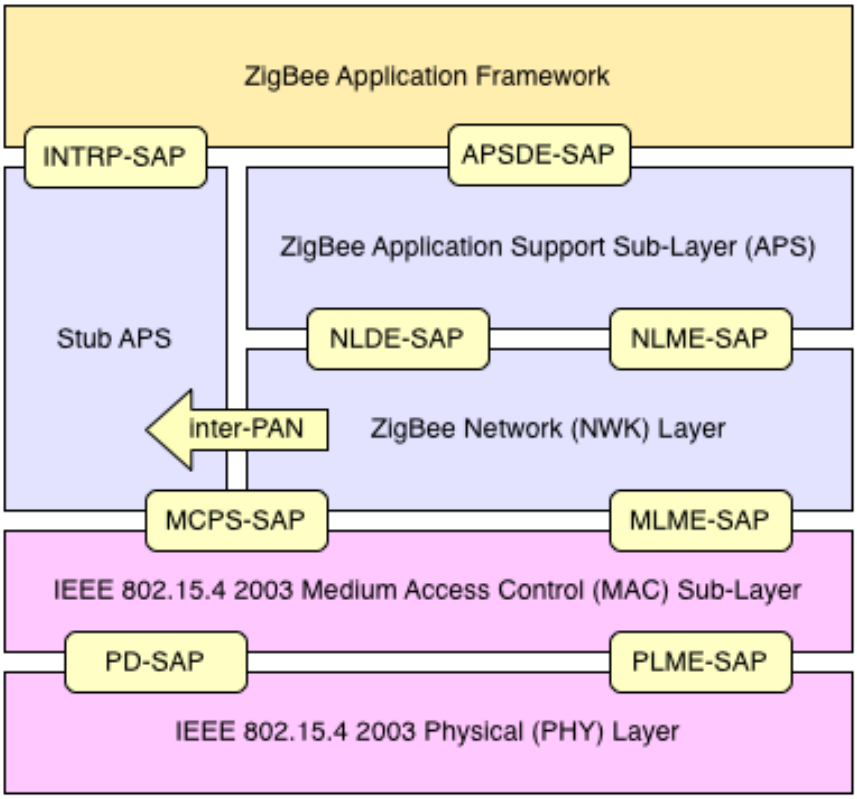


Figure B-1– ZigBee Stack with Stub APS

Inter-PAN data exchanges are handled by a special "stub" of the Application Support Sub-Layer, which is accessible through a special Service Access Point (SAP), the INTRP-SAP, parallel to the normal APSDE-SAP. The stub APS performs just enough processing to pass application data frames to the MAC for transmission and also to pass Inter-PAN application frames from the MAC to the application on receipt.

The Inter-PAN data exchange architecture does not support simultaneous execution by multiple application entities. Within a device, only one application entity shall use the Inter-PAN communications mechanisms.

The Enhanced Inter-PAN specification includes options for its operation and security. Each profile and feature that uses Inter-PAN will specify the options allowed.

B.3 Service Specification

The INTRP-SAP is a data service comprising three primitives.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

- INTRP-DATA.request - Provides a mechanism for a sending device to request transmission of an Inter-PAN message.
- INTRP-DATA.confirm - Provides a mechanism for a sending device to understand the status of a previous request to send an Inter-PAN message.
- INTRP-DATA.indication - Provides a mechanism for identifying and conveying an Inter-PAN message received from a sending device.

B.3.1 The INTRP-DATA.request Primitive

The INTRP-DATA.request primitive allows an application entity to request data transmission via the stub APS.

B.3.1.1 Semantics of the Service Primitive

The primitive interface is as follows:

INTRP-DATA.request	{
	SrcAddrMode
	DstAddrMode
	DstPANId
	DstAddress
	ProfileId
	ClusterId
	ASDULength
	ASDU
	ASDUHandle
	TxOptions
	}

Parameters of the primitive appear in Table B-1.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

3185 **Table B-1– Parameters of the INTRP-DATA. request**

Name	Type	Valid Range	Description
SrcAddrMode	Integer	0x03	The addressing mode for the source address used in this primitive. This parameter shall only reference the use of the 64-bit extended address: 0x03 = 64-bit extended address
DstAddrMode	Integer	0x01 – 0x03	The addressing mode for the destination address used in this primitive. This parameter can take one of the values from the following list: 0x01 = 16-bit group address 0x02 = 16-bit NWK address, normally the broadcast address 0xffff 0x03 = 64-bit extended address
DstPANID	16-bit PAN Id	0x0000 – 0xffff	The 16-bit PAN identifier of the entity or entities to which the ASDU is being transferred or the broadcast PANId 0xffff.
DstAddress	16-bit or 64-bit address	As specified by the AddrMode parameter	The address of the entity or entities to which the ASDU is being transferred.
ProfileId	Integer	0x0000 – 0xffff	The identifier of the application profile for which this frame is intended.
ClusterId	Integer	0x0000 – 0xffff	The identifier of the cluster, within the profile specified by the ProfileId parameter, which defines the application semantics of the ASDU.
ASDULength	Integer	0x00 – (*aMaxMACFrameSize* - 9)	The number of octets in the ASDU to be transmitted.
ASDU	Set of octets	-	The set of octets forming the ASDU to be transmitted.
ASDUHandle	Integer	0x00 – 0xff	An integer handle associated with the ASDU to be transmitted.
TxOptions	Bitmap	0000 xxxx (Where x can be 0 or 1)	The transmission options for the ASDU to be transferred. These are a bitwise OR of one or more of the following: 0x01 = Security enabled transmission 0x02 = Reserved (set to 0) 0x04 = Acknowledged transmission 0x08 = Reserved (set to 0)

3186
3187

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

3188 #### B.3.1.2 When Generated

3189 This primitive is generated by the local application entity when it wishes to address a frame
3190 to one or more peer application entities residing on neighboring devices with which it does not
3191 share a network association.

3192

3193 #### B.3.1.3 Effect on Receipt

3194 On receipt of the INTRP-DATA.request primitive by the stub APS, the stub APS will construct
3195 and transmit a frame containing the given ASDU and other parameters using the MCPS-
3196 DATA.request primitive of the MAC sub-layer, as described in sub-clause B.5.1, and, once the
3197 corresponding MCPS-DATA.confirm primitive is received, generate the INTRP-DATA.confirm
3198 primitive with a status value reflecting the status value returned by the MAC.

3199 ### B.3.2 The INTRP-DATA.confirm Primitive

3200 The INTRP-DATA.confirm primitive allows the stub APS to inform the application entity about
3201 the status of a data request.

3202 #### B.3.2.1 Semantics of the Service Primitive

3203 The primitive interface is as follows:

3204
3205
3206
3207

INTRP-DATA.confirm	{ ASDUHandle
	Status
	}

3208 Parameters of the primitive appear in Table B-2.

3209 **Table B-2– Parameters of the INTRP-DATA. confirm**

Name	Type	Valid Range	Description
ASDUHandle	Integer	0x00 – 0xff	An integer handle associated with the transmitted frame.
Status	Enumeration	Any Status value returned by the MAC	The status of the ASDU transmission corresponding to ASDUHandle as returned by the MAC.

3210

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

B.3.2.2 When Generated

This primitive is generated by the stub APS on a ZigBee device and passed to the application in response to the receipt of a MCPS-DATA.confirm primitive that is a confirmation of a previous MCPS-DATA.request issued by the stub APS.

B.3.2.3 Effect on Receipt

As a result of the receipt of this primitive, the application is informed of the results of an attempt to send a frame via the stub APS.

B.3.3 The INTRP-DATA.indication Primitive

The INTRP-DATA.indication primitive allows the stub APS to inform the next higher layer that it has received a frame that was transmitted via the stub APS on another device.

B.3.3.1 Semantics of the Service Primitive

The primitive interface is as follows:

INTRP-DATA.indication	{
	SrcAddrMode
	SrcPANId
	SrcAddress
	DstAddrMode
	DstPANId
	DstAddress
	ProfileId
	ClusterId
	ASDULength
	ASDU
	Status
	SecurityStatus
	LinkQuality
	}

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

3241 Parameters of the primitive appear in Table B-3.

3242 **Table B-3– Parameters of the INTRP-DATA.indication**

Name	Type	Valid Range	Description
SrcAddrMode	Integer	0x03	The addressing mode for the source address used in this primitive. This parameter shall only reference the use of the 64- bit extended address: 0x03 = 64-bit extended address
SrcPANId	16-bit PAN Id	0x0000 – 0xffff	The 16-bit PAN identifier of the entity from which the ASDU is being transferred.
SrcAddress	64-bit address	As specified by the SrcAddrMode parameter	The device address of the entity from which the ASDU is being transferred.
DstAddrMode	Integer	0x01 – 0x03	The addressing mode for the destination address used in this primitive. This parameter can take one of the values from the following list: 0x01 = 16-bit group address 0x02 = 16-bit NWK address, normally the broadcast address 0xffff 0x03 = 64-bit extended address
DstPANID	16-bit PAN Id	0x0000 – 0xffff	The 16-bit PAN identifier of the entity or entities to which the ASDU is being transferred or the broadcast PAN ID 0xffff.
DstAddress	16-bit or 64-bit address	As specified by the DstAddrMode parameter	The address of the entity or entities to which the ASDU is being transferred.
ProfileId	Integer	0x0000 – 0xffff	The identifier of the application profile for which this frame is intended.
ClusterId	Integer	0x0000 – 0xffff	The identifier of the cluster, within the profile specified by the ProfileId parameter, which defines the application semantics of the ASDU.
ASDULength	Integer	0x00 – (*aMaxMACFrameSize* - 9)	The number of octets in the ASDU to be transmitted.
ASDU	Set of octets	-	The set of octets forming the ASDU to be transmitted.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Status	Enumeration	SUCCESS DEFRAG_UNSUPPORTED DEFRAG_DEFERRED or any status returned from the security processing of the frame	The status of the incoming frame processing
SecurityStatus	Enumeration	UNSECURED SECURED_LINK_KEY	UNSECURED if the ASDU was without any security. SECURED_LINK_KEY if the ASDU was secured with a link key
LinkQuality	Integer	0x00 – 0xff	The link quality observed during the reception of the ASDU.

3243

3244 #### B.3.3.2 When Generated

3245 This primitive is generated and passed to the application in the event of the receipt, by the
3246 stub APS, of a MCPS-DATA.indication primitive from the MAC sub-layer, containing a frame
3247 that was generated by the stub APS of a peer ZigBee device, and that was intended for the
3248 receiving device.

3249 #### B.3.3.3 Effect on Receipt

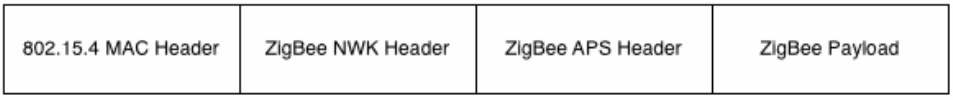
3250 Upon receipt of this primitive the application is informed of the receipt of an application
3251 frame transmitted, via the stub APS, by a peer device and intended for the receiving device.

3252 ### B.3.4 Qualifying and Testing of Inter-PAN Messages

3253 Certification and application level testing shall ensure both the sending and receiving devices
3254 correctly react and understand the INTRP-DATA.request and INTRP-DATA.indication
3255 primitives.

3256 ## B.4 Frame Formats

3257 The birds-eye view of a normal ZigBee frame is as shown in Figure B-2.



3258
3259 **Figure B-2– Normal ZigBee Frame**

3260 Briefly, the frame contains the familiar headers controlling the operation of the MAC sub-
3261 layer, the NWK layer and the APS. Following these, there is a payload, formatted as specified in
3262 [B1].

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Since most of the information contained in the NWK and APS headers is not relevant for Inter-PAN transmission, the Inter-PAN frame, shown Figure B-3, contains only a stub of the NWK header and the APS header, which provide the information required by the stub APS shown in Figure B-4 to do its job.

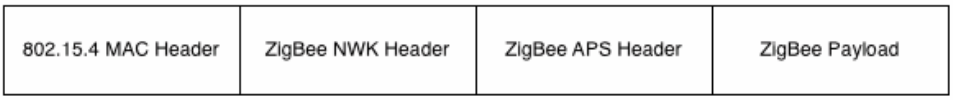


Figure B-3– Enhanced Inter-PAN ZigBee Frame

Octets: 2
NWK frame control

Figure B-4– Stub NWK Header Format

The format of the frame control field of the stub NWK header is formatted as shown in Figure B-5.

Bits: 0-1	**2-5**	**6-15**
Frame type	Protocol version	Remaining sub-fields == 0

Figure B-5– NWK Frame Control Field

The sub-fields of the NWK frame control field are as follows:

- The frame type sub-field shall have a value of 0b11, which is a reserved frame type with respect to the [B3].
- The value protocol version sub-field shall reflect the protocol version of the ZigBee stack as described in [B3].

All other sub-fields shall have a value of 0.

The format of the stub APS header is shown in Figure B-6.

Octets: 1	**0/2**	**2**	**2**	**0/1**	**0/3**	**0/14**
APS frame control	Group address	Cluster identifier	Profile identifier	APS Counter	Extended Header	Auxiliary Header
	Addressing fields					

Figure B-6– Stub APS Header Format

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

The stub APS header contains as many as seven fields totaling a maximum of 25 octets in length.

The APS frame control field shall be 1 octet in length and is identical in format to the frame control field of the general APDU frame in [B3] (see Figure B-7).

Bits: 0-1	2-3	4	5	6	7
Frame type	Delivery Mode	Reserved	Security	ACK request	Extended Header Present

Figure B-7– APS Frame Control Field

The fields of the frame control field have the following values:

- The frame type sub-field shall have a value of 0b11, which is a reserved frame type with respect to the [B3].
- The delivery mode sub-field may have a value of 0b00, indicating unicast, 0b10, indicating broadcast or 0b11 indicating group addressing.
- The Security sub-field may have the value 0 for no security or 1 for a secured frame. When the Security sub-field is 1 an Auxiliary Header shall be present.
- The ACK request sub-field shall have a value of 0 when no ACK is requested and value 1 when an ACK is requested. When an ACK is requested, the APS Counter field shall be present.
- The extended header present sub-field shall have a value of 0, indicating no extended header, or a value of 1, indicating that a message has been fragmented and that there is an extended header.

The optional Group Address shall be present if and only if the Delivery Mode field has a value of 0x0b11 and the delivery mode is 0b11 indicating a group address. If present, it shall contain the 16-bit identifier of the group to which the frame is addressed.

The Cluster Identifier field is 2 octets in length and specifies the identifier of the cluster to which the frame relates and which shall be made available for filtering and interpretation of messages at each device that takes delivery of the frame.

The Profile Identifier is two octets in length and specifies the ZigBee profile identifier for which the frame is intended and shall be used during the filtering of messages at each device that takes delivery of the frame. When the Profile Identifier is set to that of the Smart Energy profile, the Security sub-field of the APS Control field shall be set as follows. If the cluster ID is set to 0x0019 (Key Establishment), the Security sub-field shall be set to 0; otherwise it shall be set to 1.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

B.5 Frame Processing

Assuming the INTRP-SAP described above, frames transmitted using the stub APS are processed as described here.

B.5.1 Enhanced Inter-PAN Transmission

On receipt of the INTRP-DATA.request primitive, the stub APS shall construct a stub APS frame. The header of the stub APS frame shall contain a NWK and an APS frame control field as described in clause B.4, a cluster identifier field equal to the value of the ClusterId parameter of the INTRP-DATA.request and a profile identifier field equal to the value of the ProfileId parameter. If the DstAddrMode parameter of the INTRP-DATA.request has a value of 0x01, indicating group addressing, then the APS header shall also contain a group address field with a value corresponding to the value of the DstAddress parameter. The payload of the stub APS frame shall contain the data payload to be transmitted.

The stub APS frame will then be transmitted using the MCPS-DATA.request primitive of the MAC sub-layer with key primitive parameters set as follows:

- The value of the SrcAddrMode parameter of the MCPS-DATA.request shall always be set to a value of three, indicating the use of the 64-bit extended address.
- The SrcPANId parameter shall be equal to the value of the *macPANID* attribute of the MAC PIB.
- The SrcAddr parameter shall always be equal to the value of the MAC sub- layer constant *aExtendedAddress*.
- If the DstAddrMode parameter of the INTRP-DATA.request primitive has a value of 0x01, then the DstADdrMode parameter of the MCPS-DATA.request shall have a value of 0x02. Otherwise, the DstAddrMode parameter of the MCPS-DATA.request shall reflect the value of the DstAddrMode parameter of the INTRP-DATA.request.
- The DstPANId parameter shall have the value given by the DstPANID parameter of the INTRP-DATA.request primitive.
- If the DstAddrMode parameter of the INTRP-DATA.request has a value of 0x01, indicating group addressing, then the value of the DstAddr parameter of the MCPS-DATA.request shall be the broadcast address 0xffff. Otherwise, the value of the DstAddr parameter shall reflect the value of the DstAddress parameter of the INTRP-DATA.request primitive.
- The MsduLength parameter shall be the length, in octets, of the stub APS frame.
- The Msdu parameter shall be the stub APS frame itself.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

- If the transmission is a unicast packet, then the value of the TxOptions parameter shall be 0x01, indicating a request for acknowledgement. Otherwise, the TxOptions parameter shall have a value of 0x00, indicating no options.

On receipt of the MCPS-DATA.confirm primitive from the MAC sub-layer, the stub APS will invoke the transmit confirmation function with a status reflecting the status returned by the MAC.

B.5.2 Enhanced Inter-PAN Reception

On receipt of the MCPS-DATA.indication primitive from the MAC sub-layer, the receiving entity - in the case of a ZigBee device this is normally the NWK layer - shall determine whether the frame should be passed to the stub APS or processed as specified in [B3]. For a frame that is to be processed by the stub APS, the non- varying sub-fields of both the NWK frame control field and the APS frame control field must be set exactly as described above.

If the delivery mode sub-field of the APS frame control field of the stub APS header has a value of 0b11, indicating group addressing, then, if the device implements group addressing, the value of the group address field shall be checked against the NWK layer group table, and, if the received value is not present in the table, the frame shall be discarded with no further processing or action.

On receipt of a frame for processing, the stub APS shall generate an INTRP- DATA.indication with parameter values as follows:

- The value of the SrcAddrMode parameter of the INTRP-DATA.indication shall always be set to a value of three, indicating the use of the 64-bit extended address

- The value of the SrcPANId parameter shall reflect that of the SrcPANId parameter of the MCPS-DATA.indication.

- The SrcAddress parameter of the INTRP-DATA.indication shall always reflect the value of a 64-bit extended address.

- Values for the DstAddrMode parameter shall be one of the following:
  - 0x03, if the DstAddrMode parameter of the INTRP-DATA.indication has a value of 0x03.

  - 0x02, if the DstAddrMode parameter of the INTRP-DATA.indication has a value of 0x02

- The value of the DstPANId parameter of the INTRP-DATA.indication shall reflect the value of the DstPANId parameter of the MCPS-DATA.indication.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

- If the DstAddrMode parameter of the INTRP-DATA.indication has a value of 0x01, indicating group addressing, then the DstAddress parameter of the INTRP-DATA.indication shall reflect the value of the Group Address field of the stub APS header. Otherwise, the value of the DstAddress parameter of the INTRP-DATA.indication shall reflect the value of the DstAddr parameter of the MCPS-DATA.indication.

- The value of the ProfileId parameter shall be the same as the value of the Profile Identifier field of the stub APS header.

- The value of the ClusterId parameter shall be the same as the value of the Cluster Identifier field of the stub APS header.

- The ASDULength field shall contain the number of octets in the stub APS frame payload.

- The ASDU shall be the stub APS payload itself.

- The value of the LinkQuality parameter shall reflect the value of the mpduLinkQuality parameter of the MCPS-DATA.indication.

B.6 Initiating an Enhanced Inter-PAN Interaction

1. Unless Inter-PAN is being used in conjunction with Network Joining, it is assumed that there will be no need to permit joining on the network on which the required receiving device is present.

2. To find the required receiving device with which to interact, the initiating device shall send Beacon Request packets on each channel, dwelling on each channel as specified by the ZigBee PRO specification beacon response window.

3. The device shall survey all required channels and build a list of devices which have emitted a beacon that has been heard. If the initiating device has a user interface, the available devices should be displayed in order that the required receiving device can be chosen. If no user interface is available, then other method(s) shall be employed to choose a required receiving device (e.g. strongest signal).

4. The device initiates a CBKE process with the required receiving device by sending an *Initiate Key Establishment Request* command within an unsecured Enhanced Inter-PAN frame (Security sub-field set to 0).

5. The required receiving device may employ techniques to ensure that the initiating device is acceptable for such communication (e.g. by checking its EUI-64 address against a list), and refusing the transaction if it is not suitable.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

6. If the initiating device is acceptable to the required receiving device, the Enhanced Inter-PAN CBKE process will be allowed to complete, and the resultant shared APS link key shall be used to secure all further Enhanced Inter-PAN frames (Security sub-field set to 1).

B.7 Best Practices

Network Channel Manager Inter-PAN support is not specified in Annex E of the core stack specification ([B3]). New channel notifications will not be broadcast using Inter-PAN. Inter-PAN devices which do not receive the network channel change will need to perform the network discovery procedure described in B.3.4.

B.8 Security Requirements

Smart Energy profile devices supporting Enhanced Inter-PAN shall use APS security for all Inter-PAN frames except for those key establishment messages that do not require it.

In addition, devices shall verify the correct format of all SE messages received over Enhanced Inter-PAN. Any received message that does not conform to the format described in this document shall be dropped.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Annex C KEY ESTABLISHMENT CLUSTER

The candidate material in this annex, when approved, will be merged into the Foundation document of the ZigBee Cluster Library (ZCL) by the Cluster Library Development Board.

C.1 Scope and Purpose

This Annex specifies a cluster, which contains commands and attributes necessary for managing secure communication between ZigBee devices.

This Annex should be used in conjunction with the ZigBee Cluster Library, Foundation Specification (see [B1]), which gives an overview of the library and specifies the frame formats and general commands used therein.

This version is specifically for inclusion in the Smart Energy profile. The document which originates from [B4] will continue to be developed in a backward-compatible manner as a more general secure communication cluster for ZigBee applications as a whole.

C.2 General Description

C.2.1 Introduction

As previously stated, this document describes a cluster for managing secure communication in ZigBee. The cluster is for Key Establishment.

C.2.2 Security Credentials

Key Establishment requires that the device utilize pre-installed security credentials that are unique to the device. Depending on the number of cryptographic suites that the device supports, there may be multiple credentials installed. It is assumed that the device is capable of managing this and to provide the corresponding credentials based on what suite is being actively used. The mechanism for negotiating the Key Establishment suite is described in section C.3.1.1.1.

C.2.3 Network Security

The Key Establishment Cluster has been designed to be used where the underlying network security cannot be trusted. As such, no information that is confidential information will be transported.

C.2.4 Key Establishment

To allow integrity and confidentiality of data passed between devices, cryptographic schemes need to be deployed. The cryptographic scheme deployed in the ZigBee Specification for frame integrity and confidentiality is based upon a variant of the AES-CCM described in [B15] called

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

AES-CCM*. This relies on the existence of secret keying material shared between the involved devices. There are methods to distribute this secret keying material in a trusted manner. However, these methods are generally not scalable or communication may be required with a trusted key allocation party over an insecure medium. This leads to the requirement for automated key establishment schemes to overcome these problems.

Key establishment schemes can either be effected using either a key agreement scheme or a key transport scheme. The key establishment scheme described in this document uses a key agreement scheme, therefore key transport schemes will not be considered further in this document.

A key agreement scheme is where both parties contribute to the shared secret and therefore the secret keying material to be established is not sent directly; rather, information is exchanged between both parties that allows each party to derive the secret keying material. Key agreement schemes may use either symmetric key or asymmetric key (public key) techniques. The party that begins a key agreement scheme is called the initiator, and the other party is called the responder.

Key establishment using key agreement involves an initiator and a responder and four steps:

1 Establishment of a trust relationship

2 Exchange of ephemeral data

3 Use of this ephemeral data to derive secret keying material using key agreement

4 Confirmation of the secret keying material.

There are two basic types of key establishment which can be implemented:

- Symmetric Key Key Establishment
- Public Key Key Establishment

C.2.5 Symmetric Key Key Establishment

Symmetric Key Key Establishment (SKKE) is based upon establishing a link key based on a shared secret (master key). If the knowledge of the shared secret is compromised, the established link key can also be compromised. If the master key is publicly known or is set to a default value, it is known as Unprotected Key Establishment (UKE). SKKE is the key establishment method used in the ZigBee specification therefore it will not be considered any further.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

C.2.6 Public Key Key Establishment

Public Key Key Establishment (PKKE) is based upon establishing a link key based on shared static and ephemeral public keys. As the public keys do not require any secrecy, the established link key cannot be compromised by knowledge of them.

As a device's static public key is used as part of the link key creation, it can either be transported independently to the device's identity where binding between the two is assumed, or it can be transported as part of a implicit certificate signed by a Certificate Authority, which provides authentication of the binding between the device's identity and its public key as part of the key establishment process. This is called Certificate-Based Key Establishment (CBKE) and is discussed in more detail in sub-clause C.4.2.

CBKE provides the most comprehensive form of Key Establishment and therefore will be the method specified in this cluster.

The purpose of the key agreement scheme as described in this document is to produce shared secret keying material which can be subsequently used by devices using AES-CCM* the cryptographic scheme deployed in the ZigBee Specification or for any proprietary security mechanism implemented by the application.

C.2.7 General Exchange

The following diagram shows an overview of the general exchange which takes place between initiator and responder to perform key establishment.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

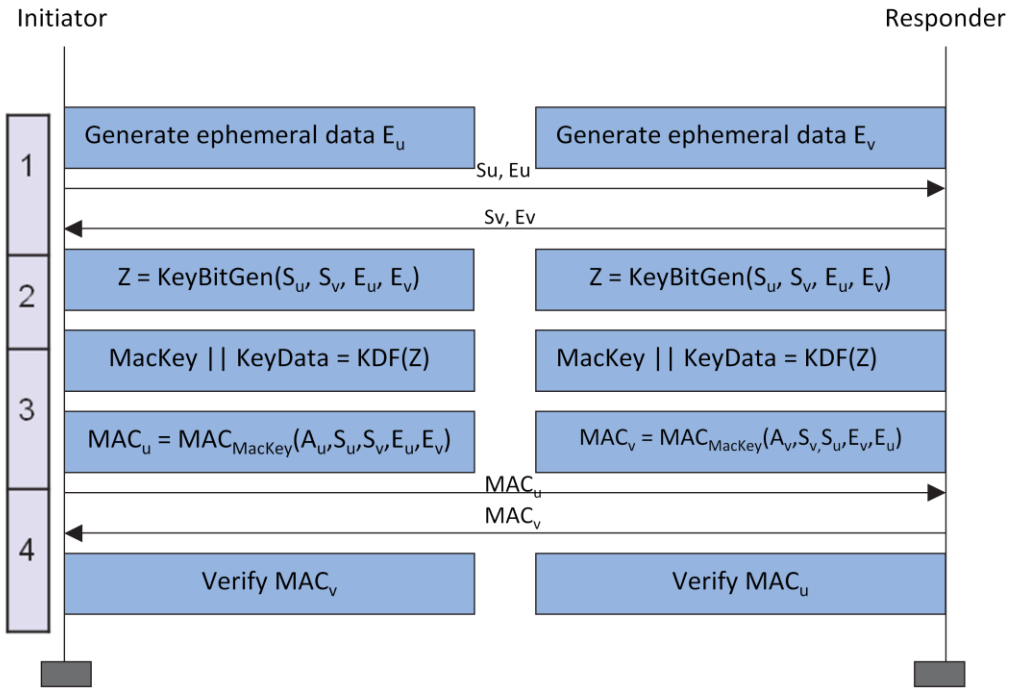


Figure C-1– Overview of General Exchange

The functions are as follows:

1 Exchange Static and Ephemeral Data

2 Generate Key Bitstream

3 Derive MAC key and Key Data

4 Confirm Key using MAC

The functions shown in the diagram (Figure C-1) depend on the Key Establishment mechanism.

C.2.7.1 Exchange Static and Ephemeral Data

Figure C-1 shows static data S_U and S_V. For PKKE schemes, this represents a combination of the 64-bit device address [B11] and the device's static public key. The identities are needed by the MAC scheme and the static public keys are needed by the key agreement scheme.

Figure C-1 also shows ephemeral data E_U and E_V. For PKKE schemes, this represents the public key of a randomly generated key pair.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

The static and ephemeral data S_U and E_U are sent to V and the static and ephemeral data S_V and E_V and are sent to U.

C.2.7.2 Generate Key Bitstream

Figure C-1 shows the KeyBitGen function for generating the key bitstream. The function's four parameters are the identifiers and the ephemeral data for both devices. This ensures the same key is generated at both ends.

For PKKE schemes, this is the ECMQV key agreement schemes specified in Section 6.2 of SEC1 [B18]. The static data S_U represents the static public key $Q_{1,U}$ of party U, the static data S_V represents the static public key $Q_{1,V}$ of party V, the ephemeral data E_U represents the ephemeral public key $Q_{2,U}$ of party U and the ephemeral data E_V represents the ephemeral public key $Q_{2,V}$ of party V.

C.2.7.3 Derive MAC Key and Key Data

Figure C-1 shows the KDF (KeyDerivation Function) for generating the MAC Key and key data. The MAC Key is used with a keyed hash message authentication function to generate a MAC and the key data is the shared secret, e.g. the link key itself required for frame protection.

For PKKE schemes, this is the key derivation function as specified in Section 3.6.1 of SEC1 [B18]. Note there is no *SharedInfo* parameter of the referenced KDF, i.e. it is a null octet string of length 0.

Figure C-1 also shows generation of the MAC using the MAC Key derived using the KDF using a message comprised of both static data S_U and S_V and ephemeral data E_U and E_V plus an additional component A which is different for initiator and responder.

For PKKE schemes, this is the MAC scheme specified in section 3.7 of SEC1 [B18]. The MAC in the reference is the keyed hash function for message authentication specified in sub-clause C.4.2.2.6 and the message M is a concatenation of the identity (the 64-bit device address [B11]) of U, the identity of V and point-compressed octet-string representations of the ephemeral public keys of parties U and V. The order of concatenation depends on whether it is the initiator or responder. The additional component A is the single octet 02_{16} for the initiator and 03_{16} for the responder.

C.2.7.4 Confirm Key Using MAC

Figure C-1 shows MACs MAC_U and MAC_V.

The MAC MAC_U is sent to V and the MAC MAC_V is sent to U. U and V both calculate the corresponding MAC and compare it with the data received.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

C.3 Cluster List

The clusters specified in this document are listed in Table C-1.

For our purposes, any device that implements the client side of this cluster may be considered the initiator of the secure communication transaction.

Table C-1– Clusters Specified for the Secure Communication Functional Domain

Cluster Name	Description
Key Establishment	Attributes and commands for establishing a shared secret between two ZigBee devices.

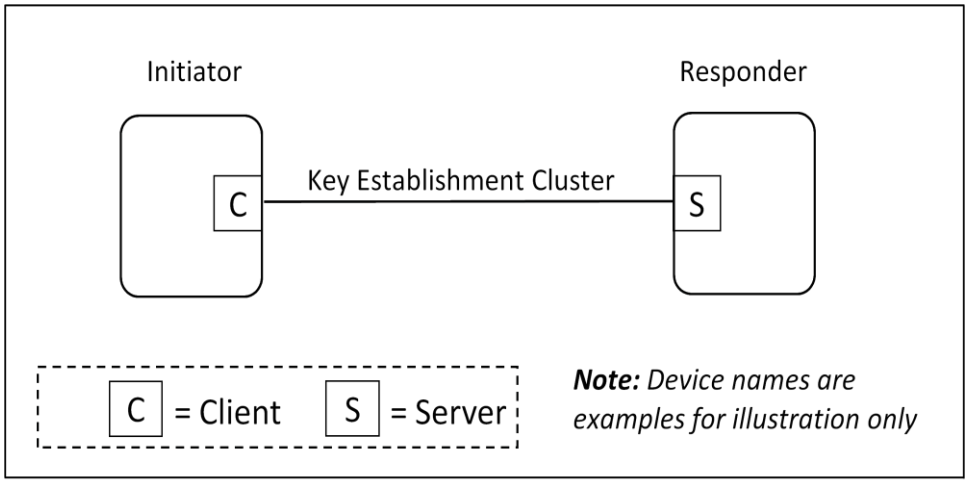


Figure C-2– Typical Usage of the Key Establishment Cluster

C.3.1 Key Establishment Cluster

C.3.1.1 Overview

This cluster provides attributes and commands to perform mutual authentication and establish keys between two ZigBee devices. Figure C-3 depicts a diagram of a successful key establishment negotiation.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

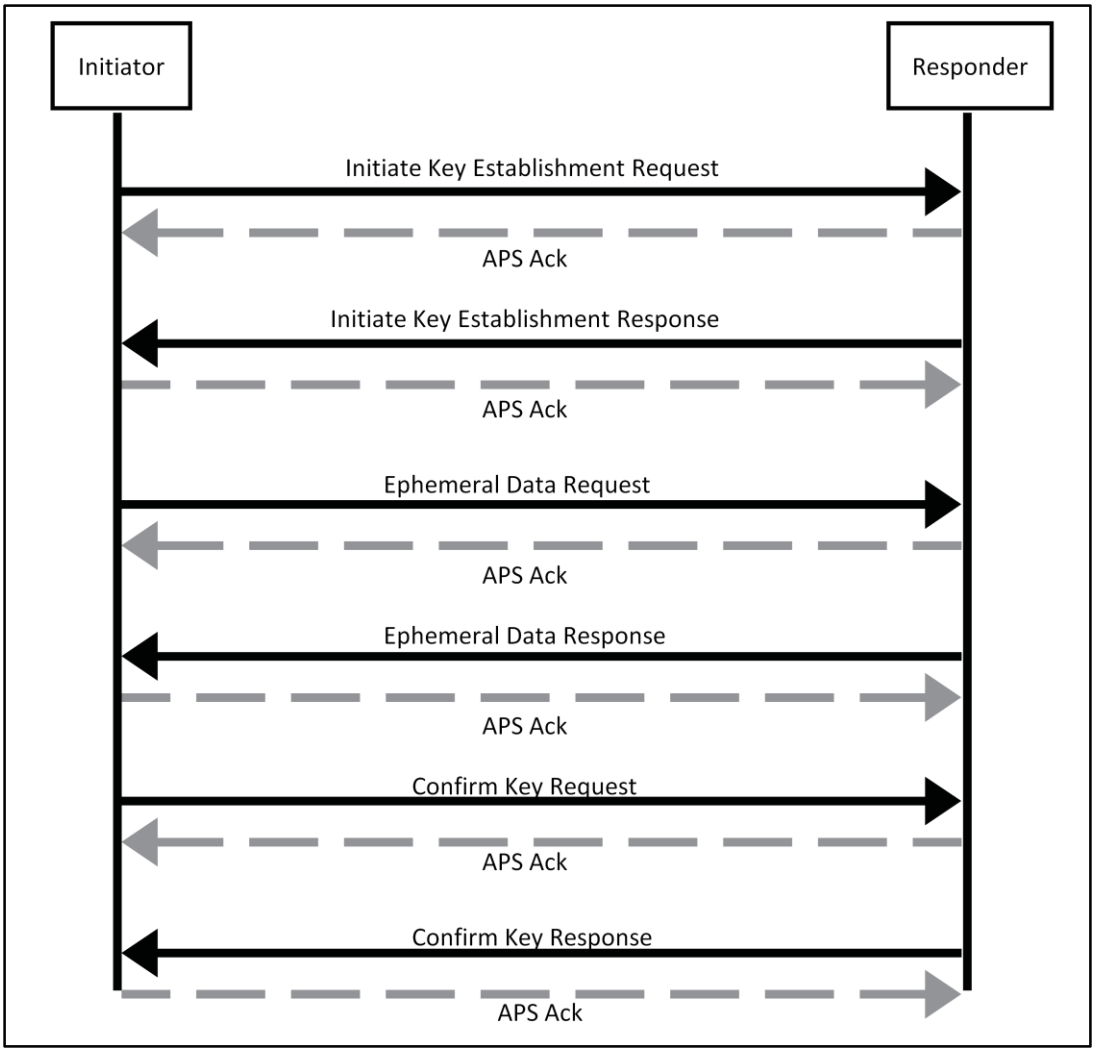


Figure C-3– Key Establishment Command Exchange

As depicted above, all Key Establishment messages should be sent with APS retries enabled. A failure to receive an ACK in a timely manner can be seen as a failure of key establishment. No Terminate Key Establishment should be sent to the partner of device that has timed out the operation.

The initiator can initiate the key establishment with any active endpoint on the responder device that supports the key establishment cluster. The endpoint can be either preconfigured or discovered, for example, by using ZDO Match-Desc-req. A link key successfully established using key establishment is valid for all endpoints on a particular device. The responder shall respond to the initiator using the source endpoint of the initiator's messages as the destination endpoint of the responder's messages.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

It is expected that the time it takes to perform the various cryptographic computations of the key establishment cluster may vary greatly based on the device. Therefore rather than set static timeouts, the *Initiate Key Establishment Request* and *Response* messages will contain approximate values for how long the device will take to generate the ephemeral data and how long the device will take to generate confirm key message.

A device performing key establishment can use this information in order to choose a reasonable timeout for its partner during those operations. The timeout should also take into consideration the time it takes for a message to traverse the network including APS retries. A minimum transmission time of 2 seconds is recommended.

For the Initiate Key Establishment Response message, it is recommended the initiator wait at least 2 seconds before timing out the operation. It is not expected that generating an Initiate Key Establishment Response will take significant time compared to generating the Ephemeral Data and Confirm Key messages.

C.3.1.1.1 Negotiating the Key Establishment Suite

Devices may support multiple cryptographic key establishment suites and therefore the client and server must agree on the suite that is to be used. Devices shall only advertise the suites that they support and have security credentials for.

The client device is expected to negotiate the key establishment suite with the server, which will be used for the rest of the key establishment exchange. The initiating device (client) may perform a Read Attribute request on the *KeyEstablishmentSuite* attribute of the server. It will then compare its local value of the attribute to the server's value to determine the common set of suites that are supported by both. The client shall choose the common suite with the highest bit value and then send the *Initiate Key Establishment Request* message using that suite. If no common suites are supported, the device shall leave the network.

C.3.1.2 Server

C.3.1.2.1 Dependencies

The Key Establishment server cluster has no dependencies.

C.3.1.2.2 Attributes

For convenience, the attributes defined in this specification are arranged into sets of related attributes; each set can contain up to 16 attributes. Attribute identifiers are encoded such that the most significant three nibbles specify the attribute set and the least significant nibble specifies the attribute within the set. The currently defined attribute sets are listed in Table C-2.

Table C-2– Key Establishment Attribute Sets

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

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

3606

3607 C.3.1.2.2.1 Information

3608 The Information attribute set contains the attributes summarized in Table C.3.

3609 **Table C-3– Key Establishment Attribute Sets**

Identifier	Name	Type	Range	Access	Default	Mandatory/ Optional
0x0000	*KeyEstablishmentSuite*	16-bit Enumeration	0x0000 - 0xFFFF	Read only	0x0000	M

3610

3611 ***C.3.1.2.2.1.1 KeyEstablishmentSuite Attribute***

3612 The *KeyEstablishmentSuite* attribute is 16-bits in length and specifies ALL the cryptographic
3613 schemes for key establishment on the device. A device shall set the corresponding bit to 1 for
3614 every cryptographic scheme that it supports. All other cryptographic schemes and reserved bits
3615 shall be set to 0.

3616 Although, for backwards compatibility, the Type cannot be changed, this 16-bit Enumeration
3617 should be treated as if it were a 16-bit BitMap.

3618 **Table C-4– Values of the *KeyEstablishmentSuite* Attribute**

Bits	Description
0	Certificate-based Key Establishment Cryptographic Suite 1 (“Crypto Suite 1”)
1	Certificate-based Key Establishment Cryptographic Suite 2 (“Crypto Suite 2”)
2-15	Reserved

3619

3620 C.3.1.2.3 **Commands Received**

3621 The server side of the key establishment cluster is capable of receiving the commands listed
3622 in Table C-5.

3623 **Table C-5– Received Command IDs for the Key Establishment Cluster Server**

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Command Identifier Field Value	Description	Mandatory/ Optional
0x00	*Initiate Key Establishment Request*	M
0x01	*Ephemeral Data Request*	M
0x02	*Confirm Key Data Request*	M
0x03	*Terminate Key Establishment*	M
0x04 – 0xFF	Reserved	

3624

3625 C.3.1.2.3.1 Initiate Key Establishment Request Command

3626 The *Initiate Key Establishment Request* command allows a device to initiate key establishment
3627 with another device. The sender shall indicate the identity information and key establishment
3628 protocol information that it wishes to use to the receiving device.

3629 ***C.3.1.2.3.1.1 Payload Format***

3630 The *Initiate Key Establishment Request* command payload shall be formatted as illustrated in
3631 Figure C-4.

Octets	2	1	1	Variable
Data Type	16-bit BitMap	Unsigned 8-bit Integer	Unsigned 8-bit Integer	Octets (non-ZCL Data Type)
Field Name	Key Establishment suite	Ephemeral Data Generate Time	Confirm Key Generate Time	Identity (IDU)

3632 **Figure C-4– *Initiate Key Establishment Request* Command Payload**

3633 **Key Establishment Suite:** This will be the type of *KeyEstablishmentSuite* that the initiator is
3634 requesting for the Key Establishment Cluster. For 'Crypto Suite 1' this will be 0x0001. For
3635 'Crypto Suite 2' this will be 0x0002. Only one suite shall be indicated in the command.

3636 **Ephemeral Data Generate Time:** This value indicates approximately how long the initiator
3637 device will take in seconds to generate the *Ephemeral Data Request* command. The valid
3638 range is 0x00 to 0xFE.
3639
3640 **Confirm Key Generate Time:** This value indicates approximately how long the initiator
3641 device will take in seconds to generate the *Confirm Key Request* command. The valid range is
3642 0x00 to 0xFE.
3643
3644 **Identity field:** The identity field shall be the block of octets containing the implicit certificate
3645 CERTU. For *KeyEstablishmentSuite* = 0x0001 ('Crypto Suite 1'), the certificate is specified in

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

sub-clause C.4.2.2. For *KeyEstablishmentSuite* = 0x0002 (‘Crypto Suite 2’) the certificate is specified in sub-clause C.4.2.3.

C.3.1.2.3.1.2 Effect on Receipt

If the device does not currently have the resources to respond to a key establishment request it shall send a *Terminate Key Establishment* command with the result value set to NO_RESOURCES and the Wait Time field shall be set to an approximation of the time that must pass before the device will have the resources to process a new Key Establishment Request.

If the receiving device does not support the cryptographic suite specified in the message, it shall send a *Terminate Key Establishment* message with the status of UNSUPPORTED_SUITE.

If the *KeyEstablishmentSuite* field of the message has more than a single bit selected in the bitmap, the receiving device shall send a *Terminate Key Establishment* message with the status of BAD_MESSAGE.

The receiving device shall extract the Issuer field of the implicit certificate received in the message. It shall then examine all locally installed certificates using the same Cryptographic suite specified in the received message and compare the Issuer field contained within the certificate to the issuer within the received certificate. If no locally installed certificates match the issuer in the received certificate, the device shall send a *Terminate Key Establishment* command with the result set to UNKNOWN_ISSUER.

If the implicit certificate received in the message is for the ‘Crypto Suite 2’ Cipher Suite, then the receiving device shall check the status of the *KeyUsage* field and, if the *Key Agreement* flag is NOT set, shall send a *Terminate Key Establishment* message with the status of INVALID_CERTIFICATE. The receiving device shall also check the *Type*, *Curve* and *Hash* fields of such a certificate, and send a *Terminate Key Establishment* message with the status of INVALID_CERTIFICATE if any of these fields contains an invalid value.

If the device accepts the request it shall send an *Initiate Key Establishment Response* command containing its own identity information. It shall set the Key Establishment suite to the same value as in the received *Initiate Key Establishment Request* message. The identity information shall correspond to the same suite as specified in the Key Establishment suite. The device should verify the certificate belongs to the address that the device is communicating with. The binding between the identity of the communicating device and its address is verifiable using an out-of-band method.

For all future server messages within the current key establishment negotiation, the Key Establishment suite value received in this message shall be utilized. If the client receives a *Terminate Key Establishment* message, or times out the operation, the key establishment suite value must be renegotiated.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

3681 C.3.1.2.3.2 Ephemeral Data Request Command

3682 The *Ephemeral Data Request* command allows a device to communicate its ephemeral data
3683 to another device and request that the device send back its own ephemeral data.

3684 ***C.3.1.2.3.2.1 Payload Format***

Octets	**Variable**
Data Type	Octets (non-ZCL Data Type)
Field Name	Ephemeral Data (QEU)

3685 **Figure C-5– *Ephemeral Data Request* Command Payload**

3686 ***C.3.1.2.3.2.2 Effect on Receipt***

3687 If the device is not currently in the middle of negotiating Key Establishment with the sending
3688 device when it receives this message, it shall send back a Terminate Key Establishment
3689 message with a result of BAD_MESSAGE. If the device is in the middle of Key Establishment
3690 with the sender but did not receive this message in response to an *Initiate Key Establishment*
3691 *Response* command, it shall send back a Terminate Key Establishment message with a result of
3692 BAD_MESSAGE. If the device can process the request it shall respond by generating its
3693 own ephemeral data and sending an *Ephemeral Data Response* command containing that value.

3694 The length of the frame shall correlate to the current key establishment suite that has been negotiated by
3695 the client and server (refer to Table C-14 for relevant sizes). If the data is shorter than the expected
3696 length according to the cryptographic suite, the responder shall send back a Terminate Key
3697 Establishment message with a result of BAD_MESSAGE.

3698 C.3.1.2.3.3 Confirm Key Request Command

3699 The *Confirm Key Request* command allows the initiator sending device to confirm the key
3700 established with the responder receiving device based on performing a cryptographic hash
3701 using part of the generated keying material and the identities and ephemeral data of both parties.

3702 ***C.3.1.2.3.3.1 Payload Format***

3703 The *Confirm KeyRequest* command payload shall be formatted as illustrated in Figure C-6.

3704

Octets	**16**
Data Type	Octets (non-ZCL Data Type)
Field Name	Secure Message Authentication Code (*MACU*)

3705 **Figure C-6– *Confirm Key Request* Command Payload**

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Secure Message Authentication Code field: The Secure Message Authentication Code field shall be the octet representation of *MACU* as specified in sub-clause C.4.2.

C.3.1.2.3.3.2 Effect on Receipt

If the device is not currently in the middle of negotiating Key Establishment with the sending device when it receives this message, it shall send back a Terminate Key Establishment message with a result of BAD_MESSAGE. If the device is in the middle of Key Establishment with the sender but did not receive this message in response to an *Ephemeral Data Response* command, it shall send back a Terminate Key Establishment message with a result of BAD_MESSAGE.

On receipt of the *Confirm Key Request* command the responder device shall compare the received MACU value with its own reconstructed version of MACU. If the two match the responder shall send back MACV by generating an appropriate *Confirm Key Response* command. If the two do not match, the responder shall send back a Terminate Key Establishment with a result of BAD KEY_CONFIRM and terminate the key establishment.

C.3.1.2.3.4 Terminate Key Establishment Command

The *Terminate Key Establishment* command may be sent by either the initiator or responder to indicate a failure in the key establishment exchange.

C.3.1.2.3.4.1 Payload Format

The *Terminate Key Establishment* command payload shall be formatted as illustrated in Figure C-7.

Octets	**1**	**1**	**2**
Data Type	8-bit Enumeration	Unsigned 8-bit Integer	16-bit BitMap
Field Name	Status Code	Wait Time	KeyEstablishmentSuite

Figure C-7– *Terminate Key Establishment* Command Payload

Status Field: The Status field shall be one of the error codes in Table C-6.

Table C-6– *Terminate Key Establishment* Command Status Field

Enumeration	**Value**	**Description**
	0x00	Reserved
UNKNOWN_ISSUER	0x01	The Issuer field within the key establishment partner's certificate is unknown to the sending device, and it has terminated the key establishment.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

BAD_KEY_CONFIRM	0x02	The device could not confirm that it shares the same key with the corresponding device and has terminated the key establishment.
BAD_MESSAGE	0x03	The device received a bad message from the corresponding device (e.g. message with bad data, an out of sequence number, or a message with a bad format) and has terminated the key establishment.
NO_RESOURCES	0x04	The device does not currently have the internal resources necessary to perform key establishment and has terminated the exchange.
UNSUPPORTED_SUITE	0x05	The device does not support the specified key establishment suite in the partner's Initiate Key Establishment message.
INVALID_CERTIFICATE	0x06	The received certificate specifies a type, curve, hash, or other parameter that is either unsupported by the device or invalid
	0x07 - 0xFF	Reserved

Wait Time: This value indicates the minimum amount of time in seconds the initiator device should wait before trying to initiate key establishment again. The valid range is 0x00 to 0xFE.

KeyEstablishmentSuite: This value will be set the value of the *KeyEstablishmentSuite* attribute. It indicates the list of key exchange methods that the device supports.

C.3.1.2.3.4.2 Effect on Receipt

On receipt of the *Terminate Key Establishment* command the device shall terminate key establishment with the sender. If the device receives a status of BAD_MESSAGE or NO_RESOURCES it shall wait at least the time specified in the Wait Time field before trying to re-initiate Key Establishment with the device.

If the device receives a status of UNSUPPORTED_SUITE it should examine the KeyEstablishmentSuite field to determine if another suite can be used that is supported by the partner device. It may re-initiate key establishment using that one of the supported suites after waiting the amount of time specified in the Wait Time field. If the device does not support any of the types in the KeyEstablishmentSuite field, it should not attempt key establishment again with that device.

If the device receives a status of UNKNOWN_ISSUER or BAD_KEY_CONFIRM the device should not attempt key establishment again with the device, as it is unlikely that another attempt will be successful.

C.3.1.2.4 **Commands Generated**

The server generates the commands detailed in sub-clause C.3.1.3.3, as well as those used for reading and writing attributes.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

C.3.1.3 Client

C.3.1.3.1 Dependencies

The Key Establishment client cluster has no dependencies.

C.3.1.3.2 Attributes

For convenience, the attributes defined in this specification are arranged into sets of related attributes; each set can contain up to 16 attributes. Attribute identifiers are encoded such that the most significant three nibbles specify the attribute set and the least significant nibble specifies the attribute within the set. The currently defined attribute sets are listed in Table C-7.

Table C-7– Key Establishment Attribute Sets

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

C.3.1.3.2.1 Information

The Information attribute set contains the attributes summarized in Table C-8.

Table C-8– Attributes of the Information Attribute Set

Identifier	Name	Type	Range	Access	Default	Mandatory/ Optional
0x0000	*KeyEstablishmentSuite*	16-bit Enumeration	0x0000 – 0xFFFF	Read only	0x0000	M

C.3.1.3.2.1.1 KeyEstablishmentSuite Attribute

The *KeyEstablishmentSuite* attribute is 16-bits in length and specifies ALL the cryptographic schemes for key establishment on the device. A device shall set the corresponding bit to 1 for every cryptographic scheme that is supports. All other cryptographic schemes and reserved bits shall be set to 0. This attribute shall be set to one of the non-reserved values listed in Table C-9.

Although, for backwards compatibility, the Type cannot be changed, this 16-bit Enumeration should be treated as if it were a 16-bit BitMap.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

3775
3776 **Table C-9– Values of the *KeyEstablishmentSuite* Attribute**

KeyEstablishmentSuite	Description
0	Certificate-based Key Establishment Cryptographic Suite 1 (“Crypto Suite 1”)
1	Certificate-based Key Establishment Cryptographic Suite 2 (“Crypto Suite 2”)
2-15	Reserved

3777

3778 C.3.1.3.3 **Commands Received**

3779 The client side of the Key Establishment cluster is capable of receiving the commands
3780 listed in Table C-10.

3781 **Table C-10– Received Command IDs for the Key Establishment Cluster Client**

Command Identifier Field Value	Description	Mandatory / Optional
0x00	*Initiate Key Establishment Response*	M
0x01	*Ephemeral Data Response*	M
0x02	*Confirm Key Data Response*	M
0x03	*Terminate Key Establishment*	M
0x04 - 0xFF	Reserved	

3782

3783 C.3.1.3.3.1 Initiate Key Establishment Response Command

3784 The *Initiate Key Establishment Response* command allows a device to respond to a device
3785 requesting the initiation of key establishment with it. The sender will transmit its identity
3786 information and key establishment protocol information to the receiving device.

3787 ***C.3.1.3.3.1.1 Payload Format***

3788 The *Initiate Key Establishment Response* command payload shall be formatted as illustrated in
3789 Figure C-8.

Octets	2	1	1	Variable
Data Type	16-bit BitMap	Unsigned 8-bit Integer	Unsigned 8-bit Integer	Octets (non-ZCL Data Type)

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Field Name	Requested Key Establishment suite	Ephemeral Data Generate Time	Confirm Key Generate Time	Identity (IDU)

Figure C-8– *Initiate Key Establishment Response* Command Payload

Requested Key Establishment Suite: This will be the type of *KeyEstablishmentSuite* that the initiator has requested be used for the key establishment exchange. The responder device shall set a single bit in the bitmask indicating that it has accepted the requested suite; all other bits shall be set to zero.

Ephemeral Data Generate Time: This value indicates approximately how long in seconds the responder device takes to generate the Ephemeral Data Response message. The valid range is 0x00 to 0xFE.

Confirm Key Generate Time: This value indicates approximately how long the responder device will take in seconds to generate the Confirm Key Response message. The valid range is 0x00 to 0xFE.

Identity field: The *Identity* field shall be the block of octets containing the implicit certificate CERTU. For *KeyEstablishmentSuite* = 0x0001 (‘Crypto Suite 1’), the certificateis specified in sub-clause C.4.2.2. For KeyEstablishmentSuite = 0x0002 (‘Crypto Suite 2’), the certificate is specified in sub-clause C.4.2.2.3.

C.3.1.3.3.1.2 Effect on Receipt

If the device is not currently in the middle of negotiating Key Establishment with the sending device when it receives this message, it shall send back a Terminate Key Establishment message with a result of BAD_MESSAGE. If the device is in the middle of Key Establishment with the sender but did not receive this message in response to an *Initiate Key Establishment Request* command, it shall send back a Terminate Key Establishment message with a result of BAD_MESSAGE.

If the receiving device does not support the key establishment suite specified in the message, it shall send a *Terminate Key Establishment* message with the status of UNSUPPORTED_SUITE.

If the *Requested Key Establishment Suite* field of the message has more than a single bit selected in the bitmap, the receiving device shall send a *Terminate Key Establishment* message with the status of BAD_MESSAGE.

On receipt of this command the device shall check the Issuer field of the device's implicit certificate. If the Issuer field does not contain a value that corresponds to a known Certificate Authority, the device shall send a *Terminate Key Establishment* command with the status value set to UNKNOWN_ISSUER. If the device does not currently have the resources to respond to a key establishment request it shall send a *Terminate Key Establishment* command with the status value set to NO_RESOURCES and the Wait Time field shall be set to an

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

approximation of the time that must pass before the device has the resources to process the request.

The receiver shall verify that the *KeyEstablishmentSuite* in the *Initiate Key Establishment Response* matches the value that was sent in the *Initiate Key Establishment Request*. If the values do not match then the device shall send a Terminate Key Establishment Request with UNSUPPORTED_SUITE.

If the implicit certificate received in the message is for the 'Crypto Suite 2' Cipher Suite, then the receiving device shall check the status of the *KeyUsage* field and, if the *Key Agreement* flag is NOT set, shall send a *Terminate Key Establishment* message with the status of INVALID_CERTIFICATE. The receiving device shall also check the *Type*, *Curve* and *Hash* fields of such a certificate, and send a *Terminate Key Establishment* message with the status of INVALID_CERTIFICATE if any of these fields contains an invalid value.

If the device accepts the response it shall send an *Ephemeral Data Request* command. The device should verify the certificate belongs to the address that the device is communicating with. The binding between the identity of the communicating device and its address is verifiable using out-of-band method.

For all future client messages within the current key establishment negotiation, the Key Establishment suite value received in this message shall be utilized. If the client receives a *Terminate Key Establishment* message, or times out the operation, the key establishment suite value must be renegotiated.

C.3.1.3.3.2 Ephemeral Data Response Command

The *Ephemeral Data Response* command allows a device to communicate its ephemeral data to another device that previously requested it.

C.3.1.3.3.2.1 Payload Format

Octets	**Variable**
Data Type	Octets (non-ZCL Data Type)
Field Name	Ephemeral Data (QEV)

Figure C-9– *Ephemeral Data Response* Command Payload

C.3.1.3.3.2.2 Effect on Receipt

If the device is not currently in the middle of negotiating Key Establishment with the sending device when it receives this message, it shall send back a Terminate Key Establishment message with a result of BAD_MESSAGE. If the device is in the middle of Key Establishment with the sender but did not receive this message in response to an *Ephemeral Data Request* command, it shall send back a Terminate Key Establishment message with a result of BAD_MESSAGE.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

The length of the frame shall correlate to the current key establishment suite that has been negotiated by the client and server (refer to Table C-14 for relevant sizes). If the length of the Ephemeral Data is shorter than the expected length according to the cryptographic suite, the responder shall send back a Terminate Key Establishment message with a result of BAD_MESSAGE.

On receipt of this command if the device can handle the request it shall perform key generation, key derivation, and MAC generation. If successful it shall generate an appropriate *Confirm Key Request* command, otherwise it shall generate a Terminate Key Establishment with a result value of NO_RESOURCES.

C.3.1.3.3.3 Confirm Key Response Command

The *Confirm Key Response* command allows the responder to verify the initiator has derived the same secret key. This is done by sending the initiator a cryptographic hash generated using the keying material and the identities and ephemeral data of both parties.

C.3.1.3.3.3.1 Payload Format

The *Confirm Key Response* command payload shall be formatted as illustrated in Figure C-10.

Octets	**16**
Data Type	Octets (non-ZCL Data Type)
Field Name	Secure Message Authentication Code (*MACV*)

Figure C-10– *Confirm Key Response* Command Payload

Secure Message Authentication Code field: The Secure Message Authentication Code field shall be the octet representation of *MACV* as specified in sub-clause C.4.2.

C.3.1.3.3.3.2 Effect on Receipt

If the device is not currently in the middle of negotiating Key Establishment with the sending device when it receives this message, it shall send back a Terminate Key Establishment message with a result of BAD_MESSAGE. If the device is in the middle of Key Establishment with the sender but did not receive this message in response to an *Confirm Key Request* command, it shall send back a Terminate Key Establishment message with a result of BAD_MESSAGE.

On receipt of the *Confirm Key Response* command the initiator device shall compare the received MACV value with its own reconstructed version of the MACV. If the two match then the initiator can consider the key establishment process to be successful. If the two do

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

not match, the initiator should send a *Terminate Key Establishment* command with a result of BAD_KEY_CONFIRM.

C.3.1.3.3.4 Terminate Key Establishment Command

The *Terminate Key Establishment* command may be sent by either the initiator or responder to indicate a failure in the key establishment exchange.

C.3.1.3.3.4.1 Payload Format

Octets	**1**	**1**	**2**
Data Type	8-bit Enumeration	Unsigned 8-bit Integer	16-bit BitMap
Field Name	Status Code	Wait Time	KeyEstablishmentSuite

Figure C-11– *Terminate Key Establishment* Command Payload

Status field: The Status field shall be one of the following error codes.

Table C-11– *Terminate Key Establishment* Command Status Field

Enumeration	**Value**	**Description**
	0x00	Reserved
UNKNOWN_ISSUER	0x01	The Issuer field within the key establishment partner's certificate is unknown to the sending device, and it has terminated the key establishment.
BAD_KEY_CONFIRM	0x02	The device could not confirm that it shares the same key with the corresponding device and has terminated the key establishment.
BAD_MESSAGE	0x03	The device received a bad message from the corresponding device (e.g. message with bad data, an out of sequence number, or a message with a bad format) and has terminated the key establishment.
NO_RESOURCES	0x04	The device does not currently have the internal resources necessary to perform key establishment and has terminated the exchange.
UNSUPPORTED_SUITE	0x05	The device does not support the specified key establishment suite in the partner's Initiate Key Establishment message.
INVALID_CERTIFICATE	0x06	The received certificate specifies a type, curve, hash, or other parameter that is either unsupported by the device or invalid
	0x07 - 0xFF	Reserved

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Wait Time: This value indicates the minimum amount of time in seconds the initiator device should wait before trying to initiate key establishment again. The valid range is 0x00 to 0xFE.

KeyEstablishmentSuite: This value will be set the value of the *KeyEstablishmentSuite* attribute. It indicates the list of key exchange methods that the device supports.

C.3.1.3.3.4.2 Effect on Receipt

On receipt of the *Terminate Key Establishment* command the device shall terminate key establishment with the sender. If the device receives a status of BAD_MESSAGE or NO_RESOURCES it shall wait at least the time specified in the Wait Time field before trying to re-initiate Key Establishment with the device.

If the device receives a status of UNKNOWN_SUITE it should examine the *KeyEstablishmentSuite* field to determine if another suite can be used that is supported by the partner device. It may re-initiate key establishment using that one of the supported suites after waiting the amount of time specified in the Wait Time field. If the device does not support any of the types in the *KeyEstablishmentSuite* field, it should not attempt key establishment again with that device.

If the device receives a status of UNKNOWN_ISSUER or *BAD_KEY_CONFIRM the device should not attempt key establishment again with the* device, as it is unlikely that another attempt will be successful.

C.3.1.3.4 Commands Generated

The client generates the commands detailed in sub-clause C.3.1.2.3, as well as those used for reading and writing attributes.

C.4 Application Implementation

C.4.1 Network Security for Smart Energy Networks

The underlying network security for Smart Energy networks is assumed to be ZigBee Standard security using pre-configured link keys.

A temporary link key for a joining device is produced by performing the cryptographic hash function on a random number assigned to the joining device (e.g. serial number) and the device identifier, which is the device's 64-bit IEEE address [B11].

The joining device's assigned random number is then conveyed to the utility via an out-of-band mechanism (e.g. telephone call, or web site registration). The utility then commissions the energy service interface (ESI) at the premises where the joining device is by installing the temporary link key at the ESI on the back channel.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

When the joining device powers up, it will also create a temporary link key as above and therefore at the time of joining both the joining device and the ESI have the same temporary link key, which can be used to transport the network key securely to the joining device.

At this point, the device will be considered joined and authenticated as far as network security is concerned. The secure communication cluster can now be invoked to replace the temporary link key with a more secure link key based on public key cryptography.

C.4.2 Certificate-Based Key Establishment

The Certificate-Based Key-Establishment (CBKE) solution uses public-key technology with digital certificates and root keys. Each device has a private key and a digital certificate that is signed by a Certificate Authority (CA).

The digital certificate includes:

- Reconstruction data for the device's public key
- The device's extended 64-bit IEEE address
- Profile specific information (e.g., the device class, network id, object type, validity date, etc.).

Certificates provide a mechanism for cryptographically binding a public key to a device's identity and characteristics.

Trust for a CBKE solution is established by provisioning a CA root key and a digital certificate to each device. A CA root key is the public key paired with the CA's private key. A CA uses its private key to sign digital certificates and the CA root key is used to verify these signatures. The trustworthiness of a public key is confirmed by verifying the CA's signature of the digital certificate. Certificates can be issued either by the device manufacturer, the device distributor, or the end customer. For example, in practical situations, the CA may be a computer (with appropriate key management software) that is kept physically secure at the end customer's facility or by a third-party.

At the end of successful completion of the CBKE protocol the following security services are offered:

- Both devices share a secret link key
- Implicit Key Authentication: Both devices know with whom they share this link key.
- Key Confirmation: Each device knows that the other device actually has computed the key correctly

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

- No Unilateral Key Control: No device has complete control over the shared link key that is established.
- Perfect Forward Secrecy: if the private key gets compromised none of future and past communications are exposed
- Known Key Security resilience: Each shared link key created per session is unique

C.4.2.1 Notation and Representation

C.4.2.1.1 Strings and String Operations

A string is a sequence of symbols over a specific set (e.g., the binary alphabet {0,1} or the set of all octets). The length of a string is the number of symbols it contains (over the same alphabet). The right-concatenation of two strings *x* and *y* of length *m* and *n* respectively (notation: *x* || *y*), is the string *z* of length *m+n* that coincides with *x* on its leftmost *m* symbols and with *y* on its rightmost *n* symbols. An octet is a bit string of length 8.

C.4.2.1.2 Integers and their Representation

Throughout this specification, the representation of integers as bit strings or octet strings shall be fixed. All integers shall be represented as binary strings in most- significant-bit first order and as octet strings in most-significant-octet first order. This representation conforms to the convention in Section 2.3 of SEC1 [B18].

C.4.2.1.3 Entities

Throughout this specification, each entity shall be a DEV and shall be uniquely identified by its 64-bit IEEE device address [B11]. The parameter *entlen* shall have the integer value 64.

C.4.2.2 Cryptographic Suite 1 Building Blocks

The following cryptographic primitives and data elements are defined for use with the CBKE ‘Crypto Suite 1’ Cipher suite protocol specified in this document.

C.4.2.2.1 Elliptic-Curve Domain Parameters

The elliptic curve domain parameters used by this Cryptographic suite shall be those for the curve “sect163k1” as specified in section 3.4.1 of SEC2 [B19].

All elliptic-curve points (and operations in this section) used by the ‘Crypto Suite 1’ Cipher Suite shall be (performed) on this curve.

C.4.2.2.2 Elliptic-Curve Point Representation

All elliptic-curve points in the Cryptographic Suite 1 shall be represented as point compressed octet strings as specified in sections 2.3.3 and 2.3.4 of SEC1 [B18]. Thus, each elliptic-curve point Cryptographic Suite 1 can be represented in 22 bytes.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

C.4.2.2.3 Elliptic-Curve Key Pair

An elliptic-curve-key pair consists of an integer *d* and a point *Q* on the curve determined by multiplying the generating point *G* of the curve by this integer (i.e., *Q=dG*) as specified in section 3.2.1 of SEC1 [B18]. Here, *Q* is called the public key, whereas *d* is called the private key; the pair (*d*, *Q*) is called the key pair. Each private key shall be represented as specified in section 2.3.7 of SEC1 [B18]. Each public key shall be represented as defined in sub-clause C.4.2.1.2 of this document.

C.4.2.2.4 ECC Implicit Certificates

The exact format of the 48-byte implicit certificate IC_U used with CBKE scheme shall be specified as follows:

$IC_U = PublicReconstrKey \| Subject \| Issuer \| ProfileAttributeData$

Where,

1. *PublicReconstrKey*: the 22-byte representation of the public-key reconstruction data BEU as specified in the implicit certificate generation protocol, which is an elliptic-curve point as specified in sub-clause C.4.2.2.2 (see SEC4 [B18]);

2. *Subject*: the 8-byte identifier of the entity *U* that is bound to the public-key reconstruction data *BEU* during execution of the implicit certificate generation protocol (i.e., the extended, 64-bit IEEE 802.15.4 address [B11] of the device that purportedly owns the private key corresponding to the public key that can be reconstructed with *PublicReconstrKey*);

3. *Issuer***:** the 8-byte identifier of the CA that creates the implicit certificate during the execution of the implicit certificate generation protocol (the so-called Certificate Authority).

4. *ProfileAttributeData***:** the 10-byte sequence of octets that can be used by a ZigBee profile for any purpose. The first two bytes of this sequence is reserved as a profile identifier, which must be defined by another ZigBee standard.

5. The string I_U as specified in Step 6 of the actions of the CA in the implicit certificate generation protocol (see section SEC4 [B22]) shall be the concatenation of the *Subject*, *Issuer*, and *ProfileAttributeData*:

   $I_U = Subject \| Issuer \| ProfileAttributeData$

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

C.4.2.2.5 Block-Cipher

The block-cipher used in this specification shall be the Advanced Encryption Standard AES-128, as specified in FIPS Pub 197 [B16]. This block-cipher has a key size that is equal to the block size, in bits, i.e., *keylen*= 128.

C.4.2.2.6 Cryptographic Hash Function

The cryptographic hash function used in this specification shall be the blockcipher based cryptographic hash function specified in Annex B.6 in [B3], with the following instantiations:

1 Each entity shall use the block-cipher E as specified in sub-clause B.1.1 in [B3].

2 All integers and octets shall be represented as specified in sub-clause C.4.2.1.

The Matyas-Meyer-Oseas hash function (specified in Annex B.6 in [B3]) has a message digest size *hashlen* that is equal to the block size, in bits, of the established blockcipher.

C.4.2.2.7 Keyed Hash Function for Message Authentication

The keyed hash message authentication code (HMAC) used in this specification shall be HMAC, as specified in the FIPS Pub 198 [B17] with the following instantiations:

1 Each entity shall use the cryptographic hash *H* function as specified in sub- clause C.4.2.2.6;

2 The block size *B* shall have the integer value 16 (this block size specifies the length of the data integrity key, in bytes, that is used by the keyed hash function, i.e., it uses a 128-bit data integrity key). This is also *MacKeyLen*, the length of *MacKey*.

3 The output size *HMAClen* of the HMAC function shall have the same integer value as the message digest parameter *hashlen* as specified in sub- clause C.4.2.2.6.

C.4.2.2.8 Derived Shared Secret

The derived shared secret *KeyData* is the output of the key establishment. *KeyData* shall have length *KeyDataLen* of 128 bits.

C.4.2.3 Cryptographic Suite 2 Building Blocks

The elliptic curve domain parameters used by this Cipher suite shall be those for the curve "sect283k1" as specified in section 3.4.1 of SEC2 [B23].

All elliptic-curve points (and operations in this section) used by the 'Crypto Suite 2' Cipher Suite shall be (performed) on this curve.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

4047 C.4.2.3.1 **Elliptic-Curve Point Representation**

4048 All elliptic-curve points in the 'Crypto Suite 2' Cipher Suite shall be represented as point
4049 compressed octet strings as specified in sections 2.3.3 and 2.3.4 of SEC1 [B18]. Thus, each
4050 elliptic-curve point can be represented in 37 bytes.

4051 C.4.2.3.2 **Elliptic-Curve Key Pair**

4052 An elliptic-curve-key pair consists of an integer *d* and a point *Q* on the curve determined by
4053 multiplying the generating point *G* of the curve by this integer (i.e., *Q=dG*) as specified in section
4054 3.2.1 of SEC1 [B18]. Here, *Q* is called the public key, whereas *d* is called the private key; the
4055 pair (*d, Q*) is called the key pair. Each private key shall be represented as specified in section
4056 2.3.7 of SEC1 [B18]. Each public key shall be represented as defined in sub-clause C.4.2.1.2 of
4057 this document.

4058 C.4.2.3.3 **ECC Implicit Certificates**

4059 The exact format of the Cryptographic Suite 2 74-byte implicit certificate ICU used with CBKE
4060 scheme follows the definitions given in SEC 4 [19] for the minimal encoding scheme (MES) and
4061 shall be specified as follows:

4062 *ICU = Type* || *SerialNo* || Curve || Hash || Issuer || ValidFrom || ValidTo || Subject || KeyUsage || *PublicReconstrKey*

4063 where

4064 **1** *Type:* is a 1-byte enumeration indicating whether the implicit certificate contains extensions.
4065 For the 'Crypto Suite 2' Cipher Suite this shall be 0x00 indicating no extensions are used;

4066 **2** *SerialNo*: is an 8-byte representation of the certificate Serial Number;

4067 **3** *Curve*: is a 1-byte elliptic curve identifier. For the 'Crypto Suite 2' Cipher Suite this shall be
4068 0x0D indicating the sect283k1 curve is used;

4069 **4** *Hash*: is a 1-byte hash identifier. For the 'Crypto Suite 2' Cipher Suite, this shall be 0x08 indicating
4070 that AES-MMO is used;

4071 **5** *Issuer***:** the 8-byte address of the CA that creates the implicit certificate during the execution of
4072 the implicit certificate generation protocol (the Certificate Authority);

4073 **6** *ValidFrom***:** the 5-byte Unix time from which the certificate is valid (this signed 40-bit
4074 integer matches that defined in SEC4 [B18]). For conversion between Unix and ZigBee
4075 time, the Zigbee Epoch (January 1, 2000) equates to 946,684,800 seconds in Unix time.
4076 *NOTE that this field is currently reserved and should be set to a default value of 0*;

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

7 *ValidTo***:** a 4-byte number giving the seconds from the *ValidFrom* time for which the certificate is considered valid. A number less than 0xFFFFFFFF gives the number in seconds while 0xFFFFFFFF indicates an infinite number of seconds;

8 *Subject*: the 8-byte identifier of the entity *U* that is bound to the public-key reconstruction data *BEU* during execution of the implicit certificate generation protocol (i.e., the extended, 64-bit IEEE 802.15.4 address [B11] of the device that purportedly owns the private key corresponding to the public key that can be reconstructed with *PublicReconstrKey*);

9 *KeyUsage*: 1-byte identifier indicating the key usage. The complete bit string is defined in SEC4 [B18], the bits relevant to the 'Crypto Suite 2' Cipher Suite are:-

Table C-12– Values of the *KeyUsage* Field

Bits	Description
0	Reserved
1	Reserved
2	Reserved
3	Key Agreement
4	Reserved
5	Reserved
6	Reserved
7	Digital Signature

For usage of the 'Crypto Suite 2' Cipher Suite for Key Establishment, bit 3 shall be set;

10 *PublicReconstrKey*: the 37-byte representation of the public-key reconstruction data BEU as specified in the implicit certificate generation protocol, which is an elliptic-curve point as specified in sub-clause C.4.2.2.2 (see SEC4 [B18]).

The specification for ICu is further summarized in the following tabular form:

Table C-13– ECC Implicit Certificate format

Bytes	Name	Description
1	Type	Type of certificate = 0, implicit no extensions
8	SerialNo	Serial Number of the certificate
1	Curve	Curve identifier (sect283k1 is 13 or byte value 0x0D)
1	Hash	Hash identifier (AES-MMO is byte value 0x08)
8	Issuer	8 byte identifier, 64-bit IEEE 802.15.4 address
5	ValidFrom	40-bit Unix time from which the certificate is valid
4	ValidTo	32-bit # of seconds from the ValidFrom time for which the certificate is considered valid (0xFFFFFFFF = infinite)
8	SubjectID	8 byte identifier, 64-bit IEEE 802.15.4 address

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

1	KeyUsage	Bit flag indicating key usage (0x88 = digital signature or key agreement allowed)
37	PublicKey	37-byte compressed public key value from which the public key of the Subject is reconstructed.

4095

4096 Note that the 74-byte certificate will necessitate the use of fragmentation with associated
4097 commands.

4098

4099 C.4.2.3.4 **Block-Cipher**

4100 Refer to section C.4.2.2.5 for definition.

4101 C.4.2.3.5 **Cryptographic Hash Function**

4102 Refer to section C.4.2.2.6 for definition.

4103 C.4.2.3.6 **Keyed Hash Function for Message Authentication**

4104 Refer to section C.4.2.2.7 for definition.

4105 C.4.2.3.7 **Derived Shared Secret**

4106 Refer to section C.4.2.2.8 for definition.
4107

4108 ### C.4.2.4 Certificate-Based Key-Establishment

4109 The CBKE method is used when the authenticity of both parties involved has not been
4110 established and where implicit authentication of both parties is required prior to key agreement.

4111 The CBKE protocol has an identical structure to the PKKE protocol, except that implicit
4112 certificates are used rather than manual certificates. The implicit certificate protocol used with
4113 CBKE shall be the implicit certificate scheme with associated implicit certificate generation
4114 scheme and implicit certificate processing transformation as specified in SEC4 [B18],
4115 with the following instantiations:

4116 **1** Each entity shall be a DEV;

4117 **2** Each entity's identifier shall be its 64-bit device address [B11]; the parameter *entlen* shall
4118 have the integer value 64;

4119 **3** Each entity shall use the cryptographic hash function as specified in sub- clause C.4.2.2.6;

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

The following additional information shall have been unambiguously established between devices operating the implicit certificate scheme:

1 Each entity shall have obtained information regarding the infrastructure that will be used for the operation of the implicit certificate scheme - including a certificate format and certificate generation and processing rules (see SEC4 [B18]);

2 Each entity shall have access to an authentic copy of the elliptic-curve public keys of one or more certificate authorities that act as CA for the implicit certificate scheme (SEC4 [B18]).

The methods by which this information is to be established are outside the scope of this standard.

The methods used during the CBKE protocol are described below. The parameters used by these methods are described in Table C-14.

Table C-14– Parameters Used by Methods of the CBKE Protocol

Parameter	Size (Octets)		Description
	'Crypto Suite 1'	**'Crypto Suite 2'**	
CERTU	48	74	The initiator device's implicit certificate used to transfer the initiator device's public key (denoted $Q_{1,U}$ in the Elliptic Curve MQV scheme in SEC1 [B18]) and the initiator device's identity.
CERTV	48	74	The responder device's implicit certificate used to transfer the responder device's public key (denoted $Q_{1,V}$ in the Elliptic Curve MQV scheme in SEC1 [B18]) and the responder device's identity.
QEU	22	37	The ephemeral public key generated by the initiator device (denoted $Q_{2,U}$ in the Elliptic Curve MQV scheme in SEC1 [B18]).
QEV	22	37	The ephemeral public key generated by the responder device (denoted Q_{2V} in the Elliptic Curve MQV scheme in SEC1 [B18]).
MACU	16	16	The secure message authentication code generated by the initiator device (where the message M is $(02_{16} \| ID_U \| ID_V \| QEU \| QEV)$ and ID_U and ID_V are the initiator and responder device entities respectively as specified in sub-clause C.4.2.2.3 and *QEU* and *QEV* are the point-compressed elliptic curve points representing the ephemeral public keys of the initiator and responder respectively as specified in sub-clause C.4.2.2.2. See also section 3.7 of SEC1 [B18]).

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

MACV	16	16	The secure message authentication code generated by the responder device (where the message M is $(03_{16} \| ID_V \| ID_U \| QEV \| QEU)$ and ID_V and ID_U are the responder and initiator device entities respectively as specified in sub-clause C.4.2.2.3 and *QEV* and *QEU* are the point-compressed elliptic curve points representing the ephemeral public keys of the responder and initiator respectively as specified in sub-clause C.4.2.2.3. See also section 3.7 of SEC1 [B18]).

4132

4133 C.4.2.4.1 **Exchange Ephemeral Data**

4134 C.4.2.4.1.1 Initiator

4135 The initiator device's implicit certificate *CERTU* and a newly generated ephemeral public key
4136 *QEU* are transferred to the responder device using the *Initiate Key Establishment* command
4137 via the Key Establishment Cluster Client.

4138 C.4.2.4.1.2 Responder

4139 The responder device's implicit certificate *CERTV* and a newly generated ephemeral
4140 public key *QEV* are transferred to the initiator device using the *Initiate Key Establishment*
4141 response command via the Key Establishment Cluster Server.

4142 C.4.2.4.2 **Validate Implicit Certificates**

4143 C.4.2.4.2.1 Initiator

4144 The initiator device's Key Establishment Cluster Client processes the *Initiate Key Establishment*
4145 response command. The initiator device examines *CERTV* (formatted as IC_V as described in sub-
4146 clause C.4.2.2.4), confirms that the *Subject* identifier is the purported owner of the certificate,
4147 and runs the certificate processing steps described in section SEC4 [B19].

4148 C.4.2.4.2.2 Responder

4149 The responder device's Key Establishment Cluster Server processes the *Initiate Key*
4150 *Establishment* command. The responder device examines *CERTU* (formatted as IC_U as described
4151 in sub-clause C.4.2.2.4), confirms that the *Subject* identifier is the purported owner of the
4152 certificate, and runs the certificate processing steps described in section SEC 4 [B19].

4153 C.4.2.4.3 **Derive Keying Material**

4154 C.4.2.4.3.1 Initiator

4155 The initiator performs the Elliptic Curve MQV scheme as specified in section 6.2 of SEC1 [B18]
4156 with the following instantiations:

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

1 The elliptic curve domain parameters shall be as specified in sub- clause C.4.2.2.1;

2 The KDF shall use the cryptographic hash function specified in sub- clause C.4.2.2.2;

3 The static public key $Q_{1,U}$ shall be the static public key of the initiator;

4 The ephemeral public key $Q_{2,U}$ shall be an ephemeral public key of the initiator generated as part of this transaction;

5 The static public key $Q_{1,V}$ shall be the static public key of the responder obtained from the responder's certificate communicated to the initiator by the responder;

6 The ephemeral public key $Q_{2,V}$ shall be based on the point-compressed octet string representation *QEV* of an ephemeral key of the responder communicated to the initiator by the responder;

7 The KDF parameter *keydatalen* shall be *MacKeyLen* + *KeyDataLen*, where *MacKeyLen* is the length of *MacKey* and *KeyDataLen* is the length of *KeyData*;

8 The parameter *SharedInfo* shall be the empty string;

The initiator device derives the keying material *MacKey* and *KeyData* from the output *K* as specified in section 3.6.1 of SEC1 [B18] by using *MacKey* as the leftmost *MacKeyLen* octets of *K* and *KeyData* as the rightmost *KeyDataLen* octets of *K*. *KeyData* is used subsequently as the shared secret and *MacKey* is used for key confirmation.

C.4.2.4.3.2 Responder

The responder performs the Elliptic Curve MQV scheme as specified in section 6.2 of SEC1 [B18] with the following instantiations:

1 The elliptic curve domain parameters shall be as specified in sub- clause C.4.2.2.1;

2 The KDF shall use the cryptographic hash function specified in sub- clause C.4.2.2.2;

3 The static public key $Q_{1,U}$ shall be the static public key of the initiator obtained from the initiator's certificate communicated to the responder by the initiator;

4 The ephemeral public key $Q_{2,U}$ shall be based on the point-compressed octet string representation *QEU* of an ephemeral key of the initiator communicated to the responder by the initiator;

5 The static public key $Q_{1,V}$ shall be the static public key of the responder;

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

6 The ephemeral public key $Q_{2,V}$ shall be an ephemeral public key of the responder generated as part of this transaction;

7 The KDF parameter *keydatalen* shall be *MacKeyLen* + *KeyDataLen*, where *MacKeyLen* is the length of *MacKey* and *KeyDataLen* is the length of *KeyData*;

8 The parameter *SharedInfo* shall be the empty string;

The responder device derives the keying material *MacKey* and *KeyData* from the output *K* as specified in section 3.6.1 of SEC1 [B18] by using *MacKey* as the leftmost *MacKeyLen* octets of *K* and *KeyData* as the rightmost *KeyDataLen* octets of *K*. *KeyData* is used subsequently as the shared secret and *MacKey* is used for key confirmation.

C.4.2.4.4 Confirm Keys

C.4.2.4.4.1 Initiator

The initiator device uses *MacKey* to compute its message authentication code *MACU* and sends it to the responder device by using the *Confirm Key* command via the Key Establishment Cluster Client.

The initiator device uses *MacKey* to confirm the authenticity of the responder by calculating *MACV* and comparing it with that sent by the responder.

C.4.2.4.4.2 Responder

The responder device uses *MacKey* to compute its message authentication code *MACV* and sends it to the initiator device by using the *Confirm Key* response command via the Key Establishment Cluster Server.

The responder device uses *MacKey* to confirm the authenticity of the initiator by calculating *MACU* and comparing it with that sent by the initiator.

C.5 Key Establishment Test Vectors for Cryptographic Suite 1

The following details the key establishment exchange data transformation and validation of test vectors for a pair of Smart Energy devices using Certificate based key exchange (CBKE) using Elliptical Curve Cryptography (ECC).

C.5.1 Preconfigured Data

Each device is expected to have been preinstalled with security information prior to initiating key establishment. The preinstalled data consists of the Certificate Authority's Public Key, a device specific certificate, and a device specific private key.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

C.5.1.1 CA Public Key

The following is the Certificate Authority's Public Key.

```
02 00 FD E8 A7 F3 D1 08
42 24 96 2A 4E 7C 54 E6
9A C3 F0 4D A6 B8
```

C.5.1.2 Responder Data

The following is the certificate for device 1. The device has an IEEE of (>)0000000000000001, and will be the responder.

```
03 04 5F DF C8 D8 5F FB
8B 39 93 CB 72 DD CA A5
5F 00 B3 E8 7D 6D 00 00
00 00 00 00 00 01 54 45
53 54 53 45 43 41 01 09
00 06 00 00 00 00 00 00
```

The certificate has the following data embedded within it:

Public Key Reconstruction Data	03 04 5F DF C8 D8 5F FB 8B 39 93 CB 72 DD CA A5 5F 00 B3 E8 7D 6D
Subject (IEEE)	00 00 00 00 00 00 00 01
Issuer	54 45 53 54 53 45 43 41
Attributes	01 09 00 06 00 00 00 00 00 00

The private key for device 1 is as follows:

```
00 b8 a9 00 fc ad eb ab
bf a3 83 b5 40 fc e9 ed
43 83 95 ea a7
```

The public key for device 1 is as follows:

```
03 02 90 a1 f5 c0 8d ad
5f 29 45 e3 35 62 0c 7a
98 fa c4 66 66 a1
```

C.5.1.3 Initiator Data

The following is the certificate for device 2. The device has an IEEE of (>)0000000000000002, and will be the initiator.

```
02 06 15 E0 7D 30 EC A2
DA D5 80 02 E6 67 D9 4B
```

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

C1 B4 22 39 83 07 00 00
00 00 00 00 00 02 54 45
53 54 53 45 43 41 01 09
00 06 00 00 00 00 00 00

The certificate has the following data embedded within it:

Public Key Reconstruction Data	02 06 15 E0 7D 30 EC A2 DA D5 80 02 E6 67 D9 4B C1 B4 22 39 83 07
Subject (IEEE)	00 00 00 00 00 00 00 02
Issuer	54 45 53 54 53 45 43 41
Attributes	01 09 00 06 00 00 00 00 00 00

The private key for device 2 is as follows:

01 E9 DD B5 58 0C F7 2E
CE 7F 21 5F 0A E5 94 E4
8D F3 E7 FE E8

The public key for device 2 is:

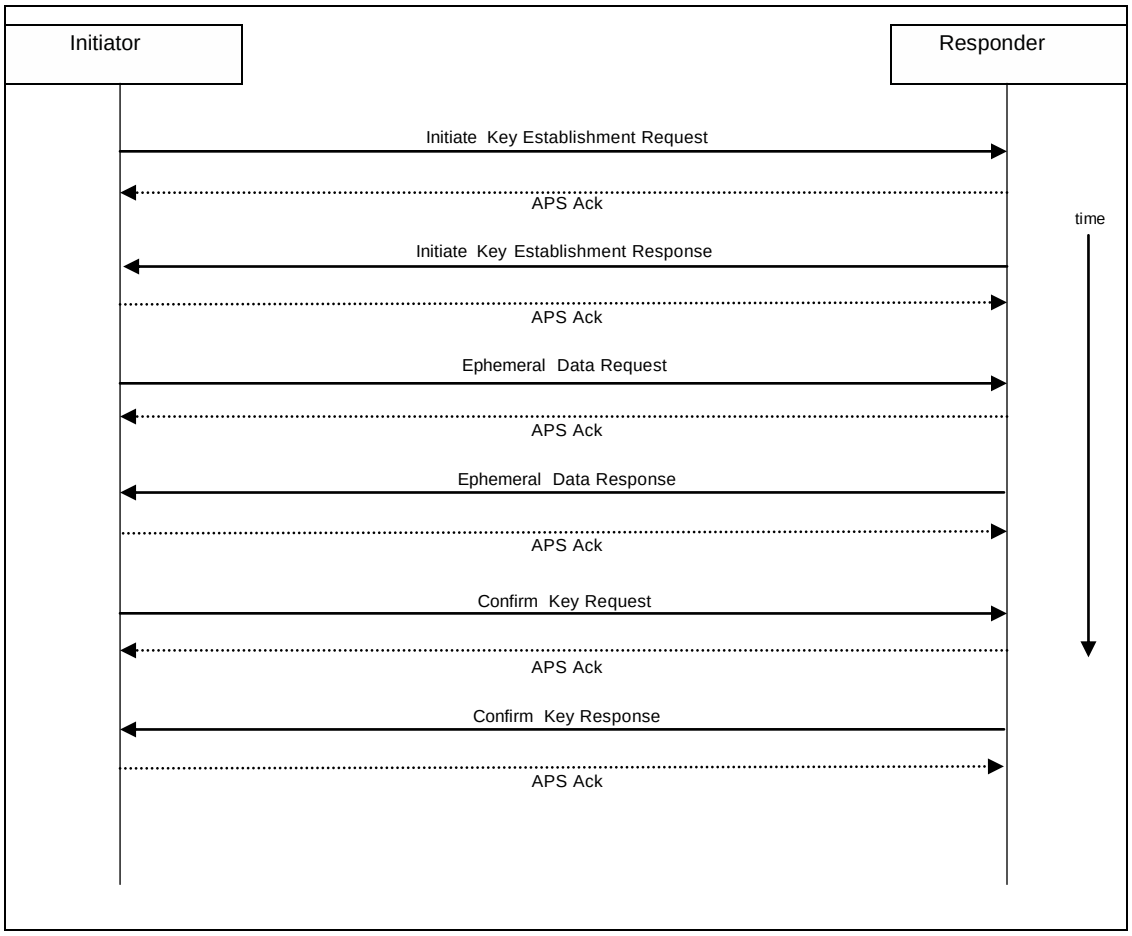
03 02 5B BA 38 D0 C7 B5
43 6B 68 DF 72 8F 09 3E
7A 1D 6C 43 7E 6D

C.5.2 Key Establishment Messages

The following is the basic flow of messages back and forth between the initiator and the responder performing key establishment using the Key Establishment Cluster.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

4262



4263 **Figure C-12– Key Establishment Command Exchange**

4264 ### C.5.2.1 Initiate Key Establishment Request

4265 The following is the APS message sent by the initiator (device 2) to the responder (device 1) for
4266 the initiate key establishment request.

4267 40 0A 00 08 09 01 0A 01
4268 01 00 00 01 00 03 06 02
4269 06 15 E0 7D 30 EC A2 DA
4270 D5 80 02 E6 67 D9 4B C1
4271 B4 22 39 83 07 00 00 00
4272 00 00 00 00 02 54 45 53
4273 54 53 45 43 41 01 09 00
4274 06 00 00 00 00 00 00

4275 **APS Header**

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Frame Control	0x40
Destination Endpoint	0x0A
Cluster Identifier	0x0800
Profile ID	0x0109
Source Endpoint	0x0A
APS Counter	0x01

ZCL Header

Frame Control	0x01	Client to Server
Sequence Number	0x00	
Command Identifier	0x00	*Initiate Key Establishment Request*
Key Establishment Suite	0x0001	ECMQV
Ephemeral Data Generate Time	0x03	
Confirm Key Generate Time	0x06	
Identity (IDU)	*	Device 2's certificate

C.5.2.2 Initiate Key Establishment Response

The following is the APS message sent by the responder (device 1) to the initiator (device 2) for the initiate key establishment response.

```
40 0A 00 08 09 01 0A 01
09 00 00 01 00 03 06 03
04 5F DF C8 D8 5F FB 8B
39 93 CB 72 DD CA A5 5F
00 B3 E8 7D 6D 00 00 00
00 00 00 00 01 54 45 53
54 53 45 43 41 01 09 00
06 00 00 00 00 00 00
```

APS Header

Frame Control	0x40
Destination Endpoint	0x0A
Cluster Identifier	0x0800
Profile ID	0x0109
Source Endpoint	0x0A
APS Counter	0x01

ZCL Header

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Frame Control	0x09	Server to Client
Sequence Number	0x00	
Command Identifier	0x00	*Initiate Key Establishment Response*
Key Establishment Suite	0x0001	ECMQV
Ephemeral Data Generate Time	0x03	
Confirm Key Generate Time	0x06	
Identity (IDV)	*	Device 1's certificate

4293

4294 ### C.5.2.3 Ephemeral Data Request

4295 The following is the APS message sent by the initiator to the responder for the ephemeral data
4296 request.

4297 `40 0A 00 08 09 01 0A 02`
4298 `01 01 01 03 00 E1 17 C8`
4299 `6D 0E 7C D1 28 B2 F3 4E`
4300 `90 76 CF F2 4A F4 6D 72`
4301 `88`

4302 **APS Header**

Frame Control	0x40
Destination Endpoint	0x0A
Cluster Identifier	0x0800
Profile ID	0x0109
Source Endpoint	0x0A
APS Counter	0x02

4303

4304 **ZCL Header**

Frame Control	0x01	Client to Server
Sequence Number	0x01	
Command Identifier	0x01	*Ephemeral Data Request*
Ephemeral Data (QEU)	`03 00 E1 17 C8 6D 0E 7C D1 28 B2 F3 4E 90 76 CF F2 4A F4 6D 72 88`	

4305

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

C.5.2.4 Ephemeral Data Response

The following is the APS message sent by the responder to the initiator for the ephemeral data response.

```
40 0A 00 08 09 01 0A 02
09 01 01 03 06 AB 52 06
22 01 D9 95 B8 B8 59 1F
3F 08 6A 3A 2E 21 4D 84
5E
```

APS Header

Frame Control	0x40
Destination Endpoint	0x0A
Cluster Identifier	0x0800
Profile ID	0x0109
Source Endpoint	0x0A
APS Counter	0x02

ZCL Header

Frame Control	0x09	Server to Client
Sequence Number	0x01	
Command Identifier	0x01	*Ephemeral Data Response*
Ephemeral Data (QEV)	03 06 AB 52 06 22 01 D9 95 B8 B8 59 1F 3F 08 6A 3A 2E 21 4D 84 5E	

C.5.2.5 Confirm Key Request

The following is the APS message sent by the initiator to the responder for the confirm key request.

```
40 0A 00 08 09 01 0A 03
01 02 02 B8 2F 1F 97 74
74 0C 32 F8 0F CF C3 92
1B 64 20
```

APS Header

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Frame Control	0x40
Destination Endpoint	0x0A
Cluster Identifier	0x0800
Profile ID	0x0109
Source Endpoint	0x0A
APS Counter	0x02

4326

4327 **ZCL Header**

Frame Control	0x01	Client to Server
Sequence Number	0x02	
Command Identifier	0x02	*Confirm Key Request*
Secure Message Authentication Code (MACU)	B8 2F 1F 97 74 74 0C 32 F8 0F CF C3 92 1B 64 20	

4328

4329 C.5.2.6 Confirm Key Response

4330 The following is the APS message sent by the responder to the initiator for the confirm key
4331 response.

4332 `40 0A 00 08 09 01 0A 03`

4333 `09 02 02 79 D5 F2 AD 1C`

4334 `31 D4 D1 EE 7C B7 19 AC`

4335 `68 3C 3C`

4336 **APS Header**

Frame Control	0x40
Destination Endpoint	0x0A
Cluster Identifier	0x0800
Profile ID	0x0109
Source Endpoint	0x0A
APS Counter	0x02

4337

4338 **ZCL Header**

Frame Control	0x09	Server to Client
Sequence Number	0x02	

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Command Identifier	0x02	*Confirm Key Response*
Secure Message Authentication Code (MACV)	79 D5 F2 AD 1C 31 D4 D1 EE 7C B7 19 AC 68 3C 3C	

4339

4340 ### C.5.3 Data Transformation

4341 The following are the various values used by the subsequent transformation.

U	Initiator
V	Responder
M(U)	Initiator Message Text (0x02)
M(V)	Responder Message Text (0x03)
ID(U)	Initiator's Identifier (IEEE address)
ID(V)	Responder's Identifier (IEEE address)
E(U)	Initiator's Ephemeral Public Key
E(V)	Responder's Ephemeral Public Key
E-P(U)	Initiator's Ephemeral Private Key
E-P(V)	Responder's Ephemeral Private Key
CA	Certificate Authority's Public Key
Cert(U)	Initiator's Certificate
Cert(V)	Responder's Certificate
Private(U)	Initiator's Private Key
Private(V)	Responder's Private Key
Shared Data	A pre-shared secret. NULL in Key Establishment.
Z	A shared secret

4342

4343 ***Note:*** *'||' stands for bitwise concatenation*

4344

4345 #### C.5.3.1 ECMQV Primitives

4346 It is assumed that an ECC library is available for creating the shared secret given the local
4347 private key, local ephemeral public & private key, remote device's certificate, remote
4348 device's ephemeral public key, and the certificate authority's public key. Further it is assumed
4349 that this library has been separately validated with a set of ECC test vectors. Those test
4350 vectors are outside the scope of this document.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

C.5.3.2 Key Derivation Function (KDF)

Once a shared secret (Z) is established, a transform is done to create a SMAC (Secure Message Authentication Code) and a shared ZigBee Key.

C.5.3.3 Initiator Transform

Upon receipt of the responder's ephemeral data response, the initiator has all the data necessary to calculate the shared secret and derive the data for the confirm key request (SMAC).

C.5.3.3.1 **Ephemeral Data**

Public Key	03 00 E1 17 C8 6D 0E 7C D1 28 B2 F3 4E 90 76 CF F2 4A F4 6D 72 88
Private Key	00 13 D3 6D E4 B1 EA 8E 22 73 9C 38 13 70 82 3F 40 4B FF 88 62

C.5.3.3.2 **Step Summary**

1. Derive the Shared Secret using the ECMQV primitives
   a. Z = ECC_GenerateSharedSecret(Private(U), E(U), E-P(U), Cert(V), E(V), CA)
2. Derive the Keying data
   a. Hash-1 = Z || 00 00 00 01 || SharedData
   b. Hash-2 = Z || 00 00 00 02 || SharedData
3. Parse KeyingData as follows
   a. MacKey = First 128 bits (Hash-1) of KeyingData
   b. KeyData = Second 128 bits (Hash-2) of KeyingData
4. Create MAC(U)
   a. MAC(U) = MAC(MacKey) { M(U) || ID(U) || ID(V) || E(U) || E(V) }
5. Send MAC(U) to V.
6. Receive MAC(V) from V.
7. Calculate MAC(V)'
   a. MAC(V) = MAC(MacKey) { M(V) || ID(V) || ID(U) || E(V) || E(U) }
8. Verify MAC(V)' is the same as MAC(V).

C.5.3.3.3 **Detailed Steps**

1. Derive the Shared Secret using the ECMQV primitives
   a. Z = ECC_GenerateSharedSecret(Private(U), E(U), E-P(U), Cert(V), E(V), CA)
      00 E0 D2 C3 CC D5 C1 06 A8 9C 4F 6C C2 6A 5F 7E

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

```
C9 DF 78 A7 BE
```

2 Derive the Keying data

a Hash-1 = Z || 00 00 00 01 || SharedData

Concatenation

```
00 E0 D2 C3 CC D5 C1 06 A8 9C 4F 6C C2 6A 5F 7E
C9 DF 78 A7 BE 00 00 00 01
```

Hash

```
90 F9 67 B2 2C 83 57 C1 0C 1C 04 78 8D E9 E8 48
```

b Hash-2 = Z || 00 00 00 02 || SharedData

Concatenation

```
00 E0 D2 C3 CC D5 C1 06 A8 9C 4F 6C C2 6A 5F 7E
C9 DF 78 A7 BE 00 00 00 02
```

Hash

```
86 D5 8A AA 99 8E 2F AE FA F9 FE F4 96 06 54 3A
```

3 Parse KeyingData as follows

a MacKey = First 128 bits (Hash-1) of KeyingData

b KeyData = Second 128 bits (Hash-2) of KeyingData

4 Create MAC(U)

a MAC(U) = MAC(MacKey) { M(U) || ID(U) || ID(V) || E(U) || E(V) }

Concatenation

```
02 00 00 00 00 00 00 00 02 00 00 00 00 00 00 00
01 03 00 E1 17 C8 6D 0E 7C D1 28 B2 F3 4E 90 76
CF F2 4A F4 6D 72 88 03 06 AB 52 06 22 01 D9 95
B8 B8 59 1F 3F 08 6A 3A 2E 21 4D 84 5E 88 00 10
```

Hash

```
B8 2F 1F 97 74 74 0C 32 F8 0F CF C3 92 1B 64 20
```

5 Send MAC(U) to V.

6 Receive MAC(V) from V.

7 Calculate MAC(V)'

a MAC(V) = MAC(MacKey) { M(V) || ID(V) || ID(U) || E(V) || E(U) }

Concatenation

```
03 00 00 00 00 00 00 00 01 00 00 00 00 00 00 00
02 03 06 AB 52 06 22 01 D9 95 B8 B8 59 1F 3F 08
6A 3A 2E 21 4D 84 5E 03 00 E1 17 C8 6D 0E 7C D1
28 B2 F3 4E 90 76 CF F2 4A F4 6D 72 88 88 00 10
```

Hash

```
79 D5 F2 AD 1C 31 D4 D1 EE 7C B7 19 AC 68 3C 3C
```

8 Verify MAC(V)' is the same as MAC(V).

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

C.5.3.4 Responder Transform

Upon receipt of the initiator's confirm key request, the responder has all the data necessary to calculate the shared secret, validate the initiator's confirm key message, and derive the data for the confirm key response (SMAC).

C.5.3.4.1 **Ephemeral Data**

Public Key	03 06 AB 52 06 22 01 D9 95 B8 B8 59 1F 3F 08 6A 3A 2E 21 4D 84 5E
Private Key	03 D4 8C 72 10 DD BC C4 FB 2E 5E 7A 0A A1 6A 0D B8 95 40 82 0B

C.5.3.4.2 **Step Summary**

1 Derive the Shared Secret using the ECMQV primitives
- a Z = ECC_GenerateSharedSecret(Private(V), E(V), E-P(V), Cert(U), E(U), CA)

2 Derive the Keying data
- a Hash-1 = Z || 00 00 00 01 || SharedData
- b Hash-2 = Z || 00 00 00 02 || SharedData

3 Parse KeyingData as follows
- a MacKey = First 128 bits (Hash-1) of KeyingData
- b KeyData = Second 128 bits (Hash-2) of KeyingData

4 Create MAC(V)
- a MAC(V) = MAC(MacKey) { M(V) || ID(V) || ID(U) || E(V) || E(U) }

5 Calculate MAC(U)'
- a MAC(U) = MAC(MacKey) { M(U) || ID(U) || ID(V) || E(U) || E(V) }

6 Verify MAC(U)' is the same as MAC(U).

7 Send MAC(V) to U.

C.5.3.4.3 **Detailed Steps**

1 Derive the Shared Secret using the ECMQV primitives
- a Z = ECC_GenerateSharedSecret(Private(U), E(U), E-P(U), Cert(V), E(V), CA)

  00 E0 D2 C3 CC D5 C1 06 A8 9C 4F 6C C2 6A 5F 7E C9 DF 78 A7 BE

2 Derive the Keying data
- a Hash-1 = Z || 00 00 00 01 || SharedData

  Concatenation

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

```
00 E0 D2 C3 CC D5 C1 06 A8 9C 4F 6C C2 6A 5F 7E
C9 DF 78 A7 BE 00 00 00 01
```

Hash

```
90 F9 67 B2 2C 83 57 C1 0C 1C 04 78 8D E9 E8 48
```

b. Hash-2 = Z || 00 00 00 02 || SharedData

Concatenation

```
00 E0 D2 C3 CC D5 C1 06 A8 9C 4F 6C C2 6A 5F 7E
C9 DF 78 A7 BE 00 00 00 02
```

Hash

```
86 D5 8A AA 99 8E 2F AE FA F9 FE F4 96 06 54 3A
```

3 Parse KeyingData as follows

a. MacKey = First 128 bits (Hash-1) of KeyingData

b. KeyData = Second 128 bits (Hash-2) of KeyingData

4 Create MAC(V)

a. MAC(V) = MAC(MacKey) { M(V) || ID(V) || ID(U) || E(V) || E(U) }

Concatenation

```
03 00 00 00 00 00 00 00 01 00 00 00 00 00 00 00
02 03 06 AB 52 06 22 01 D9 95 B8 B8 59 1F 3F 08
6A 3A 2E 21 4D 84 5E 03 00 E1 17 C8 6D 0E 7C D1
28 B2 F3 4E 90 76 CF F2 4A F4 6D 72 88 88 00 10
```

Hash

```
79 D5 F2 AD 1C 31 D4 D1 EE 7C B7 19 AC 68 3C 3C
```

5 Calculate MAC(V)'

a. MAC(U) = MAC(MacKey) { M(U) || ID(U) || ID(V) || E(U) || E(V) }

Concatenation

```
02 00 00 00 00 00 00 00 02 00 00 00 00 00 00 00
01 03 00 E1 17 C8 6D 0E 7C D1 28 B2 F3 4E 90 76
CF F2 4A F4 6D 72 88 03 06 AB 52 06 22 01 D9 95
B8 B8 59 1F 3F 08 6A 3A 2E 21 4D 84 5E 88 00 10
```

Hash

```
B8 2F 1F 97 74 74 0C 32 F8 0F CF C3 92 1B 64 20
```

6 Verify MAC(V)' is the same as MAC(V).

7 Send MAC(V) to U.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

C.6 Key Establishment Test Vectors for Cryptographic Suite 2

The following details the key establishment exchange data transformation and validation of test vectors for a pair of Smart Energy devices using Certificate based key exchange (CBKE) using Elliptical Curve Cryptography (ECC).

C.6.1 Preconfigured Data

Each device is expected to have been preinstalled with security information prior to initiating key establishment. The preinstalled data consists of the Certificate Authority's Public Key, a device specific certificate, and a device specific private key.

C.6.1.1 CA Public Key

The following is the Certificate Authority's Public Key:

```
02 07 A4 45 02 2D 9F 39   f4 9B DC 38 38 00 26 A2
7A 9E 0A 17 99 31 3A B2   8C 5C 1A 1C 6B 60 51 54
DB 1D FF 67 52
```

C.6.1.2 Responder Data

The following is the certificate for device 1. The device has an EUI-64 address of 0A:0B:0C:0D:0E:0F:10:11, and will be the responder.

Certificate:

```
00 26 22 A5 05 E8 93 8F   27 0D 08 11 12 13 14 15
16 17 18 00 52 92 A3 5B   FF FF FF FF 0A 0B 0C 0D
0E 0F 10 11 88 03 03 B4   E9 DC 54 3A 64 33 3C 98
23 08 02 2B 54 E6 7E 2F   15 F5 32 55 1B 0A 11 E2
E2 C1 C1 D3 09 7A 43 24   E7 ED
```

The certificate has the following data embedded within it:

Certificate Type	00
Certificate Serial No:	26 22 A5 05 E8 93 8F 27
Curve:	0D (sect283k1)
Hash:	08 (zigbee-aes-mmo)
IssuerID:	11 12 13 14 15 16 17 18
ValidFrom:	00 52 92 A3 5B
ValidUntil:	FF FF FF FF
SubjectID:	0A 0B 0C 0D 0E 0F 10 11
KeyUsage	88
PublicKeyReconstructionPoint	03 03 B4 E9 DC 54 3A 64 33 3C 98 23 08 02 2B 54 E6 7E 2F 15 F5 32 55 1B 0A 11 E2 E2 C1 C1 D3 09 7A 43 24 E7 ED

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

The private key for device 1 is as follows:

```
01 51 CD 0D BC B8 04 74   BF 7A C9 FE EB E3 9C 7A
32 A6 35 18 93 8F CA 97   54 AA E1 32 BC 9C 73 BE
94 A7 E1 BE
```

The public key for device 1 is as follows:

```
02 02 F4 FA 2A 30 40 43   3C 68 20 29 9D 18 2A 10
42 E4 14 04 E3 37 C5 7F   47 71 6B 42 DF AF 97 0F
15 80 A0 4C 9B
```

C.6.1.3 Initiator Data

The following is the certificate for device 2. The device has an EUI-64 address of 0A:0B:0C:0D:0E:0F:10:12, and will be the initiator.

Certificate:

```
00 84 A9 33 B3 7F 01 8D   EC 0D 08 11 12 13 14 15
16 17 18 00 52 92 A3 8A   FF FF FF FF 0A 0B 0C 0D
0E 0F 10 12 88 03 07 62   77 E2 F7 E2 25 2B 16 A0
E9 2B 6E 87 71 BB 3F 20   79 46 CB D4 A4 5D 9A 9D
F6 ED AB 8C 79 6A 48 E8   9D EC
```

The certificate has the following data embedded within it:

Certificate Type	00
Certificate Serial No:	84 A9 33 B3 7F 01 8D EC
Curve:	0D (sect283k1)
Hash:	08 (zigbee-aes-mmo)
IssuerID:	11 12 13 14 15 16 17 18
ValidFrom:	00 52 92 A3 8A
ValidUntil:	FF FF FF FF
SubjectID:	0A 0B 0C 0D 0E 0F 10 12
KeyUsage	88
PublicKeyReconstructionPoint	03 07 62 77 E2 F7 E2 25 2B 16 A0 E9 2B 6E 87 71 BB 3F 20 79 46 CB D4 A4 5D 9A 9D F6 ED AB 8C 79 6A 48 E8 9D EC

The private key for device 2 is as follows:

```
00 F2 56 1A DB 39 EF 49   C1 D6 2E F5 18 6C 6E 0C
15 8A 5A 45 BF CE 38 66   09 31 AC C3 69 45 92 D5
AC DE 90 06
```

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

The public key for device 2 is as follows:

```
03 03 0E 56 F7 AD E8 66   E7 63 72 76 4B A2 0A 9F
F1 FE 4C AE 52 2F 94 83   9E 70 F2 AD FC 1C A3 E9
7F 4D DC AF 2E
```

C.6.2 Key Establishment Messages

The following is the basic flow of messages back and forth between the initiator and the responder performing key establishment using the Key Establishment Cluster.

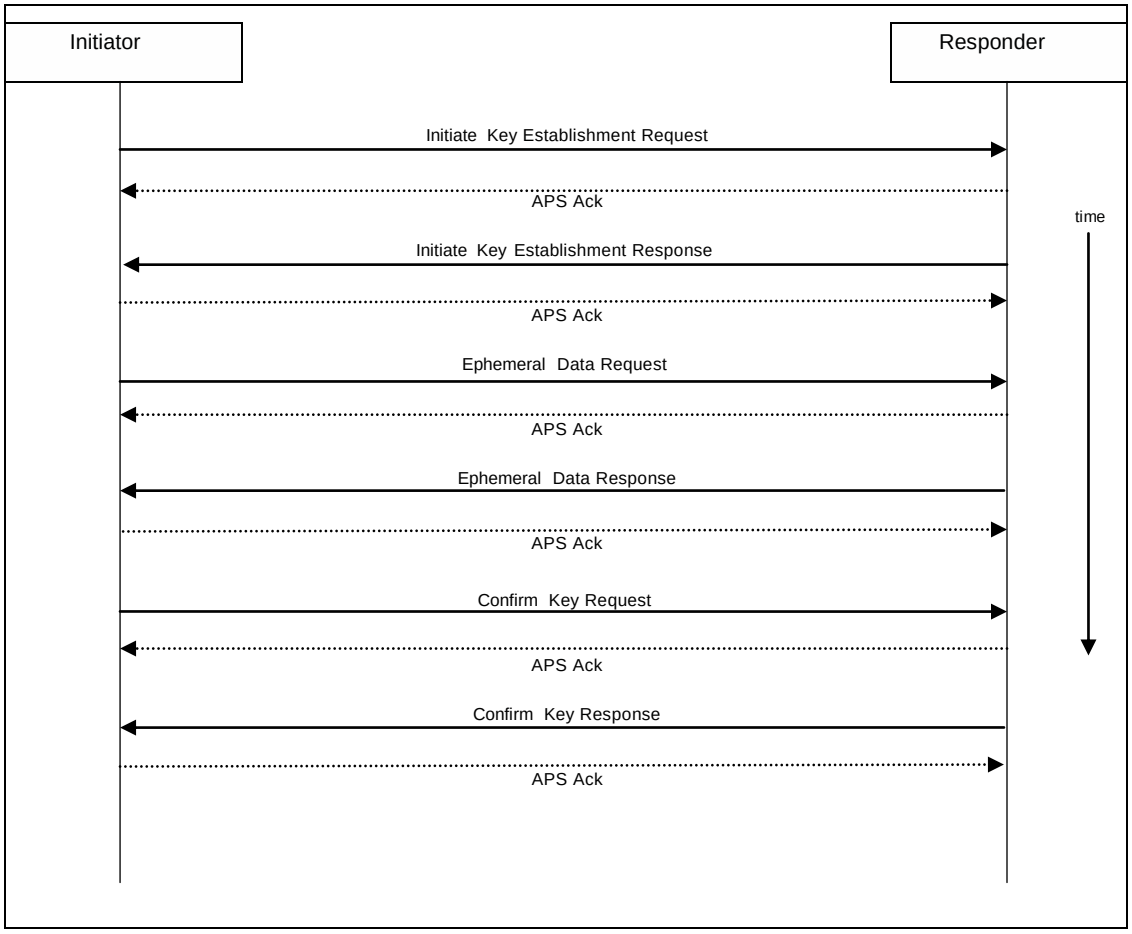


Figure C-13– Key Establishment Command Exchange

C.6.2.1 Initiate Key Establishment Request

The following is the APS message sent by the initiator (device 2) to the responder (device 1) for the initiate key establishment request.

```
40 0A 00 08 09 01 0A 01   01 00 00 02 00 03 06 00
84 A9 33 B3 7F 01 8D EC   0D 08 11 12 13 14 15 16
17 18 00 52 92 A3 8A FF   FF FF FF 0A 0B 0C 0D 0E
```

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

```
0F 10 12 88 03 07 62 77   E2 F7 E2 25 2B 16 A0 E9
2B 6E 87 71 BB 3F 20 79   46 CB D4 A4 5D 9A 9D F6
ED AB 8C 79 6A 48 E8 9D   EC
```

APS Header

Frame Control	0x40
Destination Endpoint	0x0A
Cluster Identifier	0x0800
Profile ID	0x0109
Source Endpoint	0x0A
APS Counter	0x01

ZCL Header

Frame Control	0x01	Client to Server
Sequence Number	0x00	
Command Identifier	0x00	*Initiate Key Establishment Request*
Requested Security Suite	0x0002	CBKE-ECMQV-V2
Ephemeral Data Generate Time	0x03	
Confirm Key Generate Time	0x06	
Identity (IDU)	*	Device 2's certificate

C.6.2.2 Initiate Key Establishment Response

The following is the APS message sent by the responder (device 1) to the initiator (device 2) for the initiate key establishment response.

```
40 0A 00 08 09 01 0A 01   09 00 00 02 00 03 06 00
26 22 A5 05 E8 93 8F 27   0D 08 11 12 13 14 15 16
17 18 00 52 92 A3 5B FF   FF FF FF 0A 0B 0C 0D 0E
0F 10 11 88 03 03 B4 E9   DC 54 3A 64 33 3C 98 23
08 02 2B 54 E6 7E 2F 15   F5 32 55 1B 0A 11 E2 E2
C1 C1 D3 09 7A 43 24 E7   ED
```

APS Header

Frame Control	0x40
Destination Endpoint	0x0A
Cluster Identifier	0x0800
Profile ID	0x0109
Source Endpoint	0x0A
APS Counter	0x01

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

ZCL Header

Frame Control	0x09	Server to Client
Sequence Number	0x00	
Command Identifier	0x00	*Initiate Key Establishment Response*
Accepted Security Suite	0x0002	CBKE-ECMQV-V2
Ephemeral Data Generate Time	0x03	
Confirm Key Generate Time	0x06	
Identity (IDV)	*	Device 1's certificate

C.6.2.3 Ephemeral Data Request

The following is the APS message sent by the initiator to the responder for the ephemeral data request.

```
40 0A 00 08 09 01 0A 02   01 01 01 03 05 F3 39 4E
15 68 06 60 EE CA A3 67   88 D9 B6 F3 12 B9 71 CE
2C 96 17 57 0B F7 DF CD   21 C9 72 01 77 62 C3 32
```

APS Header

Frame Control	0x40
Destination Endpoint	0x0A
Cluster Identifier	0x0800
Profile ID	0x0109
Source Endpoint	0x0A
APS Counter	0x02

ZCL Header

Frame Control	0x01	Client to Server
Sequence Number	0x01	
Command Identifier	0x01	*Ephemeral Data Request*
Ephemeral Data (QEU)	03 05 F3 39 4E 15 68 06 60 EE CA A3 67 88 D9 B6 F3 12 B9 71 CE 2C 96 17 57 0B F7 DF CD 21 C9 72 01 77 62 C3 32	

C.6.2.4 Ephemeral Data Response

The following is the APS message sent by the responder to the initiator for the ephemeral data response.

```
40 0A 00 08 09 01 0A 02   09 01 01 03 00 9A 51 31
```

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

```
CF 5B 92 A0 16 37 8C 0F   7F 28 4E CD 47 F9 40 10
F8 75 D4 3B F1 E9 A6 54   74 AD BF C6 36 96 A9 30
```

APS Header

Frame Control	0x40
Destination Endpoint	0x0A
Cluster Identifier	0x0800
Profile ID	0x0109
Source Endpoint	0x0A
APS Counter	0x02

ZCL Header

Frame Control	0x09	Server to Client
Sequence Number	0x01	
Command Identifier	0x01	*Ephemeral Data Response*
Ephemeral Data (QEV)	03 00 9A 51 31 CF 5B 92 A0 16 37 8C 0F 7F 28 4E CD 47 F9 40 10 F8 75 D4 3B F1 E9 A6 54 74 AD BF C6 36 96 A9 30	

C.6.2.5 Confirm Key Request

The following is the APS message sent by the initiator to the responder for the confirm key request.

```
40 0A 00 08 09 01 0A 03
01 02 02 BF 7E 1A 26 D4
EF 70 38 B5 68 13 E4 65
A1 31 C9
```

APS Header

Frame Control	0x40
Destination Endpoint	0x0A
Cluster Identifier	0x0800
Profile ID	0x0109
Source Endpoint	0x0A
APS Counter	0x03

ZCL Header

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Frame Control	0x01	Client to Server
Sequence Number	0x02	
Command Identifier	0x02	*Confirm Key Request*
Secure Message Authentication Code (MACU)	BF 7E 1A 26 D4 EF 70 38 B5 68 13 E4 65 A1 31 C9	

4598

4599 ### C.6.2.6 Confirm Key Response

4600 The following is the APS message sent by the responder to the initiator for the confirm key
4601 response.

4602 40 0A 00 08 09 01 0A 03
4603 09 02 02 C5 B4 32 A9 99
4604 5A 09 2F 44 49 F8 36 13
4605 93 00 64

4606 **APS Header**

Frame Control	0x40
Destination Endpoint	0x0A
Cluster Identifier	0x0800
Profile ID	0x0109
Source Endpoint	0x0A
APS Counter	0x03

4607

4608 **ZCL Header**

Frame Control	0x09	Server to Client
Sequence Number	0x02	
Command Identifier	0x02	*Confirm Key Response*
Secure Message Authentication Code (MACV)	C5 B4 32 A9 99 5A 09 2F 44 49 F8 36 13 93 00 64	

4609

4610 ## C.6.3 Data Transformation

4611 The following are the various values used by the subsequent transformation.

U	Initiator
V	Responder
M(U)	Initiator Message Text (0x02)
M(V)	Responder Message Text (0x03)
ID(U)	Initiator's Identifier (IEEE address)

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

ID(V)	Responder's Identifier (IEEE address)
E(U)	Initiator's Ephemeral Public Key
E(V)	Responder's Ephemeral Public Key
E-P(U)	Initiator's Ephemeral Private Key
E-P(V)	Responder's Ephemeral Private Key
CA	Certificate Authority's Public Key
Cert(U)	Initiator's Certificate
Cert(V)	Responder's Certificate
Private(U)	Initiator's Private Key
Private(V)	Responder's Private Key
Shared Data	A pre-shared secret. NULL in Key Establishment.
Z	A shared secret

Note: *'||' stands for bitwise concatenation*

C.6.3.1 ECMQV Primitives

It is assumed that an ECC library is available for creating the shared secret given the local private key, local ephemeral public & private key, remote device's certificate, remote device's ephemeral public key, and the certificate authority's public key. Further it is assumed that this library has been separately validated with a set of ECC test vectors. Those test vectors are outside the scope of this document.

C.6.3.2 Key Derivation Function (KDF)

Once a shared secret (Z) is established, a transform is done to create a SMAC (Secure Message Authentication Code) and a shared ZigBee Key.

C.6.3.3 Initiator Transform

Upon receipt of the responder's ephemeral data response, the initiator has all the data necessary to calculate the shared secret and derive the data for the confirm key request (SMAC).

C.6.3.3.1 **Ephemeral Data**

Public Key	03 05 F3 39 4E 15 68 06 60 EE CA A3 67 88 D9 B6 F3 12 B9 71 CE 2C 96 17 57 0B F7 DF CD 21 C9 72 01 77 62 C3 32
Private Key	00 13 D3 6D E4 B1 EA 8E 22 73 9C 38 13 70 82 3F 40 4B FF 88 62 B5 21 FE CA 98 71 FB 36 91 84 6D 36 13 04 B4

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

C.6.3.3.2 **Step Summary**

1 Derive the Shared Secret using the ECMQV primitives
- a Z = ECC_GenerateSharedSecret(Private(U), E(U), E-P(U), Cert(V), E(V), CA)

2 Derive the Keying data
- a Hash-1 = Z || 00 00 00 01 || SharedData
- b Hash-2 = Z || 00 00 00 02 || SharedData

3 Parse KeyingData as follows
- a MacKey = First 128 bits (Hash-1) of KeyingData
- b KeyData = Second 128 bits (Hash-2) of KeyingData

4 Create MAC(U)
- a MAC(U) = MAC(MacKey) { M(U) || ID(U) || ID(V) || E(U) || E(V) }

5 Send MAC(U) to V.

6 Receive MAC(V) from V.

7 Calculate MAC(V)'
- a MAC(V) = MAC(MacKey) { M(V) || ID(V) || ID(U) || E(V) || E(U) }

8 Verify MAC(V)' is the same as MAC(V).

C.6.3.3.3 **Detailed Steps**

1 Derive the Shared Secret using the ECMQV primitives
- a Z = ECC_GenerateSharedSecret(Private(U), E(U), E-P(U), Cert(V), E(V), CA)

```
04 F7 72 4A 9A 77 B2 1D   27 47 CC EF 68 A4 57 E4
52 46 C4 BE 9F 66 FD 94   25 22 7B CB 2C C5 18 0E
A9 CC CB 9A
```

2 Derive the Keying data
- a Hash-1 = Z || 00 00 00 01 || SharedData

Concatenation

```
04 F7 72 4A 9A 77 B2 1D   27 47 CC EF 68 A4 57 E4
52 46 C4 BE 9F 66 FD 94   25 22 7B CB 2C C5 18 0E
A9 CC CB 9A 00 00 00 01
```

Hash

```
ED 38 0A 00 29 66 00 FB   6B 89 30 25 DE 5F D1 37
```

- b Hash-2 = Z || 00 00 00 02 || SharedData

Concatenation

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

```
04 F7 72 4A 9A 77 B2 1D   27 47 CC EF 68 A4 57 E4
52 46 C4 BE 9F 66 FD 94   25 22 7B CB 2C C5 18 0E
A9 CC CB 9A 00 00 00 02
```

Hash

```
AA 46 89 C7 0B E0 FA F0   C9 BE 53 4A BD 9F 4C DC
```

3 Parse KeyingData as follows

a MacKey = First 128 bits (Hash-1) of KeyingData

b KeyData = Second 128 bits (Hash-2) of KeyingData

4 Create MAC(U)

a MAC(U) = MAC(MacKey) { M(U) || ID(U) || ID(V) || E(U) || E(V) }

Concatenation

```
02 0a 0b 0c 0d 0e 0f 10   12 0a 0b 0c 0d 0e 0f 10
11 03 05 F3 39 4E 15 68   06 60 EE CA A3 67 88 D9
B6 F3 12 B9 71 CE 2C 96   17 57 0B F7 DF CD 21 C9
72 01 77 62 C3 32 03 00   9A 51 31 CF 5B 92 A0 16
37 8C 0F 7F 28 4E CD 47   F9 40 10 F8 75 D4 3B F1
E9 A6 54 74 AD BF C6 36   96 A9 30
```

Hash

```
BF 7E 1A 26 D4 EF 70 38   B5 68 13 E4 65 A1 31 C9
```

5 Send MAC(U) to V.

6 Receive MAC(V) from V.

7 Calculate MAC(V)'

a MAC(V) = MAC(MacKey) { M(V) || ID(V) || ID(U) || E(V) || E(U) }

Concatenation

```
03 0a 0b 0c 0d 0e 0f 10   11 0a 0b 0c 0d 0e 0f 10
12 03 00 9A 51 31 CF 5B   92 A0 16 37 8C 0F 7F 28
4E CD 47 F9 40 10 F8 75   D4 3B F1 E9 A6 54 74 AD
BF C6 36 96 A9 30 03 05   F3 39 4E 15 68 06 60 EE
CA A3 67 88 D9 B6 F3 12   B9 71 CE 2C 96 17 57 0B
F7 DF CD 21 C9 72 01 77   62 C3 32
```

Hash

```
C5 B4 32 A9 99 5A 09 2F   44 49 F8 36 13 93 00 64
```

8 Verify MAC(V)' is the same as MAC(V).

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

C.6.3.4 Responder Transform

Upon receipt of the initiator's confirm key request, the responder has all the data necessary to calculate the shared secret, validate the initiator's confirm key message, and derive the data for the confirm key response (SMAC).

C.6.3.4.1 **Ephemeral Data**

Public Key	03 00 9A 51 31 CF 5B 92 A0 16 37 8C 0F 7F 28 4E CD 47 F9 40 10 F8 75 D4 3B F1 E9 A6 54 74 AD BF C6 36 96 A9 30
Private Key	03 D4 8C 72 10 DD BC C4 FB 2E 5E 7A 0A A1 6A 0D B8 95 40 82 0B 8D C0 91 AB 52 1E A8 24 AF E1 17 CA DE 99 5B

C.6.3.4.2 **Step Summary**

1 Derive the Shared Secret using the ECMQV primitives
- a Z = ECC_GenerateSharedSecret(Private(V), E(V), E-P(V), Cert(U), E(U), CA)

2 Derive the Keying data
- a Hash-1 = Z || 00 00 00 01 || SharedData
- b Hash-2 = Z || 00 00 00 02 || SharedData

3 Parse KeyingData as follows
- a MacKey = First 128 bits (Hash-1) of KeyingData
- b KeyData = Second 128 bits (Hash-2) of KeyingData

4 Create MAC(V)
- a MAC(V) = MAC(MacKey) { M(V) || ID(V) || ID(U) || E(V) || E(U) }

5 Calculate MAC(U)'
- a MAC(U) = MAC(MacKey) { M(U) || ID(U) || ID(V) || E(U) || E(V) }

6 Verify MAC(U)' is the same as MAC(U).

7 Send MAC(V) to U

C.6.3.4.3 **Detailed Steps**

1 Derive the Shared Secret using the ECMQV primitives
- a Z = ECC_GenerateSharedSecret(Private(V), E(V), E-P(V), Cert(U), E(U), CA)

```
04 F7 72 4A 9A 77 B2 1D   27 47 CC EF 68 A4 57 E4
52 46 C4 BE 9F 66 FD 94   25 22 7B CB 2C C5 18 0E
A9 CC CB 9A
```

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

2 Derive the Keying data

a Hash-1 = Z || 00 00 00 01 || SharedData

Concatenation

```
04 F7 72 4A 9A 77 B2 1D   27 47 CC EF 68 A4 57 E4
52 46 C4 BE 9F 66 FD 94   25 22 7B CB 2C C5 18 0E
A9 CC CB 9A 00 00 00 01
```

Hash

```
ED 38 0A 00 29 66 00 FB   6B 89 30 25 DE 5F D1 37
```

b Hash-2 = Z || 00 00 00 02 || SharedData

Concatenation

```
04 F7 72 4A 9A 77 B2 1D   27 47 CC EF 68 A4 57 E4
52 46 C4 BE 9F 66 FD 94   25 22 7B CB 2C C5 18 0E
A9 CC CB 9A 00 00 00 02
```

Hash

```
AA 46 89 C7 0B E0 FA F0   C9 BE 53 4A BD 9F 4C DC
```

3 Parse KeyingData as follows

a MacKey = First 128 bits (Hash-1) of KeyingData

b KeyData = Second 128 bits (Hash-2) of KeyingData

4 Create MAC(V)

a MAC(V) = MAC(MacKey) { M(V) || ID(V) || ID(U) || E(V) || E(U) }

Concatenation

```
03 0a 0b 0c 0d 0e 0f 10   11 0a 0b 0c 0d 0e 0f 10
12 03 00 9A 51 31 CF 5B   92 A0 16 37 8C 0F 7F 28
4E CD 47 F9 40 10 F8 75   D4 3B F1 E9 A6 54 74 AD
BF C6 36 96 A9 30 03 05   F3 39 4E 15 68 06 60 EE
CA A3 67 88 D9 B6 F3 12   B9 71 CE 2C 96 17 57 0B
F7 DF CD 21 C9 72 01 77   62 C3 32
```

Hash

```
C5 B4 32 A9 99 5A 09 2F   44 49 F8 36 13 93 00 64
```

5 Calculate MAC(U)'

a MAC(U) = MAC(MacKey) { M(U) || ID(U) || ID(V) || E(U) || E(V) }

Concatenation

```
02 0a 0b 0c 0d 0e 0f 10   12 0a 0b 0c 0d 0e 0f 10
11 03 05 F3 39 4E 15 68   06 60 EE CA A3 67 88 D9
B6 F3 12 B9 71 CE 2C 96   17 57 0B F7 DF CD 21 C9
72 01 77 62 C3 32 03 00   9A 51 31 CF 5B 92 A0 16
37 8C 0F 7F 28 4E CD 47   F9 40 10 F8 75 D4 3B F1
E9 A6 54 74 AD BF C6 36   96 A9 30
```

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Hash
BF 7E 1A 26 D4 EF 70 38 B5 68 13 E4 65 A1 31 C9

6 Verify MAC(U)' is the same as MAC(U).

7 Send MAC(V) to U

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Annex D SMART ENERGY CLUSTER DESCRIPTIONS

The candidate material in this annex describing the Smart Energy Clusters, when approved, will be merged into the Foundation document of the ZigBee Cluster Library (ZCL) by the Cluster Library Development Board.

D.1 Annex Guidelines

D.1.1 Client/Server Model Information

The ZigBee Cluster Library Specification is used as the guiding reference for defining the rule set in defining the Client/Server model for the Smart Energy Profile. Please note the following items influence the further refinement of that definition:

- Attributes can be defined for both Client and Server side clusters. Attributes can be used to understand current state of activities within a device, enhancing both the diagnostic and maintenance of devices or the processes supported by that device.

- The ESI device acts as the transition point from upstream Wide Area Network (and subsequent upstream systems) to the ZigBee network. Because of this responsibility, in some of the clusters it acts as a proxy for the upstream systems. In situations in which the proxy condition occurs, plus where attributes are defined or commands (transactions) are initiated on both client/ server sides, the ESI will be by default labeled as the Server side in the cluster descriptions.

D.1.2 Interpretation of Reserved Field Values or Bitmaps

To support backwards compatibility, devices should ignore any values or bit settings for any reserved field values. If the field is necessary for interpretation or in conjunction with other fields the whole message can be ignored.

To enable future growth and ensure backwards compatibility, any existing devices which encounter any fields applied after the end of a command shall treat them as reserved fields. The future addition of fields applied after the end of defined cluster commands are reserved solely for ZigBee specifications, Manufacturers shall not add fields after the end of commands.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

D.2 Demand Response and Load Control Cluster

D.2.1 Overview

This cluster provides an interface to the functionality of Smart Energy Demand Response and Load Control. Devices targeted by this cluster include thermostats and devices that support load control.

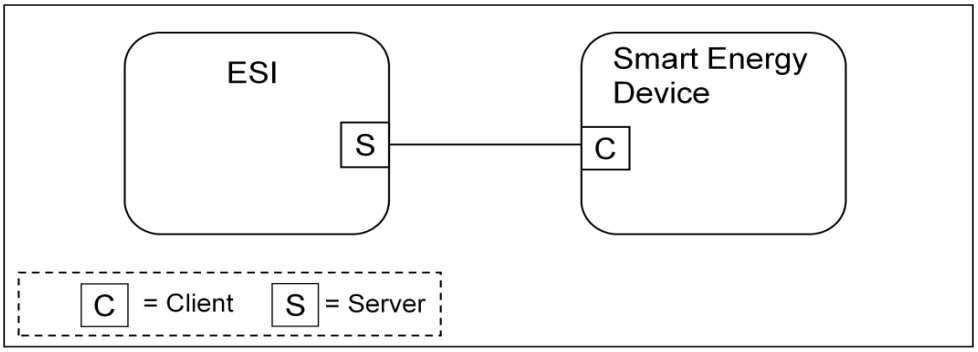


Figure D-1– Demand Response/Load Control Cluster Client Server Example

Please note the ESI is defined as the Server due to its role in acting as the proxy for upstream demand response/load control management systems and subsequent data stores.

D.2.2 Server

By default the ESI will be labeled as the Server side in the cluster descriptions, being able to initiate load control commands to other devices in the network.

D.2.2.1 Dependencies

A server device shall be capable of storing at least two load control events.

Events carried using this cluster include a timestamp with the assumption that target devices maintain a real-time clock. Devices can acquire and synchronize their internal clocks with the ESI as described in sub-clause 5.12.1.1.

If a device does not support a real-time clock, it is assumed the device will ignore all values within the Time field except the “Start Now” value.

Additionally, for devices without a real-time clock, it is assumed those devices will utilize a method (i.e. ticks, countdowns, etc.) to approximate the correct duration period.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

4824 ### D.2.2.2 Attributes

4825 There are no attributes for the Demand Response and Load Control Cluster server.

4826 ### D.2.2.3 Commands Generated

4827 The command IDs generated by the Demand Response and Load Control cluster server are
4828 listed in Table D-1.

4829 **Table D-1– Command IDs for the Demand Response and Load Control Server**

Command Identifier Field Value	Description	Mandatory/ Optional
0x00	*Load Control Event*	M
0x01	*Cancel Load Control Event*	M
0x02	*Cancel All Load Control Events*	M
0x03 – 0xff	Reserved	

4830

4831 #### D.2.2.3.1 **Load Control Event Command**

4832 ##### D.2.2.3.1.1 Payload Format

4833 The *Load Control Event* command payload shall be formatted as illustrated in Figure D-2.

Octets	4	2	1	4	2	1	1
Data Type	Unsigned 32-bit integer	16-bit BitMap	Unsigned 8-bit integer	UTC Time	Unsigned 16-bit integer	Unsigned 8-bit integer	Unsigned 8-bit integer
Field Name	Issuer Event ID (M)	Device Class (M)	Utility Enrollment Group (M)	Start Time (M)	Duration In Minutes (M)	Criticalit y Level (M)	Cooling Temperature Offset (O)

4834

Octets	1	2	2	1	1	1
Data Type	Unsigned 8-bit integer	Signed 16-bit integer	Signed 16-bit integer	Signed 8-bit integer	Unsigned 8-bit integer	8-bit BitMap
Field Name	Heating Temperature Offset (O)	Cooling Temperature Set Point (O)	Heating Temperat ure Set Point (O)	Average Load Adjustment Percentage (O)	Duty Cycle (O)	Event Control (M)

4835 **Figure D-2– *Load Control Event* Command Payload**

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

4836 ***Note:*** *M = Mandatory field, O = Optional field.* ***All fields must be present in the payload.*** *Optional fields will be*
4837 *marked with specific values to indicate they are not being used.*

4838 ***D.2.2.3.1.1.1 Payload Details***

4839 **Issuer Event ID (mandatory):** Unique identifier generated by the Energy provider. The value of
4840 this field allows matching of Event reports with a specific Demand Response and Load Control
4841 event. The expected value contained in this field shall be a unique number managed by upstream
4842 systems or a UTC based time stamp (UTCTime data type) identifying when the Load Control Event
4843 was issued.

4844 **Device Class (mandatory):** Bit encoded field representing the Device Class to apply the
4845 current Load Control Event. Each bit, if set individually or in combination, indicates the class
4846 device(s) needing to participate in the event. (Note that the participating device may be
4847 different than the controlling device. For instance, a thermostat may act on behalf of an HVAC
4848 compressor or furnace and/or Strip Heat/Baseboard Heater and should take action on their
4849 behalf, as the thermostat itself is not subject to load shed but controls devices that are subject to
4850 load shed.) The encoding of this field is in Table D-2:

4851 **Table D-2– Device Class Field BitMap/Encoding**

Bit	Description
0	HVAC Compressor or Furnace
1	Strip Heaters/Baseboard Heaters
2	Water Heater
3	Pool Pump/Spa/Jacuzzi
4	Smart Appliances
5	Irrigation Pump
6	Managed Commercial & Industrial (C&I) loads
7	Simple misc. (Residential On/Off) loads
8	Exterior Lighting
9	Interior Lighting
10	Electric Vehicle
11	Generation Systems
12 to 15	Reserved

4852
4853 Device manufacturers shall recognize the Device Class or set of Devices Classes that
4854 corresponds to its functionality. For example, a thermostat (PCT) may react when Bit 0 is set
4855 since it controls the HVAC and/or furnace. Another example is a device that acts like an EMS
4856 where it controls exterior lights, interior lights, and simple misc. load control devices. In this
4857 case the EMS would react when Bits 7, 8, or 9 are set individually or in combination.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Utility Enrollment Group (mandatory): The Utility Enrollment Group field can be used in conjunction with the Device Class bits. It provides a mechanism to direct Load Control Events to groups of Devices. Example, by assigning two different groups relating to either Demand Response programs or geographic areas, Load Control Events can be further directed for a sub-set of Device Classes (i.e. Device Class Bit 0 and Utility Enrollment Group #1 vs. Device Class Bit0 and Utility Enrollment Group #2). 0x00 addresses all groups, and values 0x01 to 0xFF address individual groups that match. Please refer to sub-clause D.2.3.2.1 for further details.

If the Device Class and/or Utility Enrollment Group fields don't apply to your End Device, the *Load Control Event* command shall be ignored by either dropping the message and not replying at all or by sending back a Default Response message with a SUCCESS status code.

Start Time (mandatory): UTC Timestamp representing when the event is scheduled to start. A start time of 0x00000000 is a special time denoting "now." If the device would send an event with a Start Time of now, adjust the Duration In Minutes field to correspond to the remainder of the event.

Duration In Minutes (mandatory): Duration of this event in number of minutes. Maximum value is 1440 (one day).

Criticality Level (mandatory): This field defines the level of criticality of this event. The action taken by load control devices for an event can be solely based on this value, or combination with other Load Control Event fields supported by this device. For example, additional fields such as Average Load Adjustment Percentage, Duty Cycle, Cooling Temperature Offset, Heating Temperature Offset, Cooling Temperature Set Point or Heating Temperature Set Point can be used in combination with the Criticality level. Criticality levels are listed in Table D-3.

Table D-3– Criticality Levels

Criticality Level	Level Description	Participation
0	Reserved	
1	Green	Voluntary
2	1	Voluntary
3	2	Voluntary
4	3	Voluntary
5	4	Voluntary
6	5	Voluntary
7	Emergency	Mandatory
8	Planned Outage	Mandatory

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

9	Service Disconnect	Mandatory
0x0A to 0x0F	Utility Defined	Utility Defined
0x10 to 0xFF	Reserved	

The criticality level 0x0 and 0x10 to 0xFF are reserved for future profile changes and not used.

"Green" event, level 0x01, may be used to denote that the energy delivered uses an abnormal amount from non-"green" sources. Participation in this event is voluntary.

The criticality levels 0x02 through 0x06 (Levels 1 through 5) indicate progressively increasing levels of load reduction are being requested by the utility. Participation in these events is voluntary.

The criticality level 0x07 is used to indicate an "Emergency" event. Participation in this event is mandatory, as defined by the utility. The expected response to this event is termination of all non-essential energy use, as defined by the utility. Exceptions to participation in this event type must be managed by the utility.

The criticality level 0x08 is used to indicate a "Planned Outage" event. Participation in this event is mandatory, as defined by the utility. The expected response to this event is termination of delivery of all non-essential energy, as defined by the utility. Exceptions to participation in this event type must be managed by the utility.

The criticality level 0x09 is used to indicate a "Service Disconnect" event. Participation in this event is mandatory, as defined by the utility. The expected response to this event is termination of delivery of all non-essential energy, as defined by the utility. Exceptions to participation in this event type must be managed by the utility.

Levels 0x0A to 0x0F are available for Utility Defined criticality levels.

Cooling Temperature Offset (optional): Requested offset to apply to the normal cooling setpoint at the time of the start of the event in + 0.1 ºC.

Heating Temperature Offset (optional): Requested offset to apply to the normal heating setpoint at the time of the start of the event in + 0.1 ºC.

The Cooling and Heating Temperature Offsets represent a temperature change (Delta Temperature) that will be applied to both the associated heating and cooling set points. The temperature offsets (Delta Temperatures) will be calculated per the Local Temperature in the Thermostat. The calculated temperature will be interpreted as the number of degrees to be added to the cooling set point and subtracted from the heating set point. Sequential demand response events are not cumulative. The Offset shall be applied to the normal setpoint.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Each offset represents the temperature offset (Delta Temperature) in degrees Celsius, as follows: Delta Temperature Offset / 10 = delta temperature in degrees Celsius. Where 0.00°C <= temperature <= 25.4 ºC, corresponding to a Temperature in the range 0x00 to 0x0FE. The maximum resolution this format allowed is 0.1 ºC.

A DeltaTemperature of 0xFF indicates that the temperature offset is not used.

If a temperature offset is sent that causes the heating or cooling temperature set point to exceed the limit boundaries that are programmed into the thermostat, the thermostat should respond by setting the temperature at the limit.

Cooling Temperature Set Point (optional): Requested cooling set point in 0.01 degrees Celsius.

Heating Temperature Set Point (optional): Requested heating set point in 0.01 degrees Celsius.

Cooling and heating temperature set points will be defined and calculated per the *LocalTemperature* attribute in the Thermostat Cluster [B1].

These fields represent the temperature in degrees Celsius, as follows:

Cooling Temperature Set Point / 100 = temperature in degrees Celsius

Where -273.15°C <= temperature <= 327.67°C, corresponding to a Cooling and/or Heating Temperature Set Point in the range 0x954d to 0x7fff.

The maximum resolution this format allows is 0.01°C.

A Cooling or Heating Temperature Set Point of 0x8000 indicates that the temperature set point is not used.

If a temperature is sent that exceeds the temperature limit boundaries that are programmed into the thermostat, the thermostat should respond by setting the temperature at the limit.

The thermostat shall not use a Cooling or Heating Temperature Set Point that causes the device to use more energy than the normal setting.

When both a Temperature Offset and a Temperature Set Point are provided, the thermostat may use either as defined by the device manufacturer. The thermostat should use the setting that provides the lowest energy consumption.

Average Load Adjustment Percentage (optional): Defines a maximum energy usage limit as a percentage of the client implementations specific average energy usage. The load adjustment

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

4943 percentage is added to 100% creating a percentage limit applied to the client implementations
4944 specific average energy usage. A -10% load adjustment percentage will establish an energy
4945 usage limit equal to 90% of the client implementations specific average energy usage. Each
4946 load adjustment percentage is referenced to the client implementations specific average energy
4947 usage. There are no cumulative effects.

4948 The range of this field is -100 to +100 with a resolution of 1 percent. A -100% value equals a
4949 total load shed. A 0% value will limit the energy usage to the client implementation's specific
4950 average energy usage. A +100% value will limit the energy usage to double the client
4951 implementation's specific average energy usage.

4952 A value of 0x80 indicates the field is not used. All other values are reserved for future use.

4953 **Duty Cycle (optional): Defines** the maximum On state duty cycle as a percentage of time.
4954 Example, if the value is 80, the device would be in an "on state" for 80% of the time for the
4955 duration of the event. Range of the value is 0 to 100. A value of 0xFF indicates the field is not
4956 used. All other values are reserved for future use.

4957 Duty cycle control is a device specific issue and shall be managed by the device manufacturer. It
4958 is expected that the duty cycle of the device under control will span the shortest practical time
4959 period in accordance with the nature of the device under control and the intent of the request
4960 for demand reduction. For typical Device Classes, three minutes7 for each 10% of duty cycle is
4961 recommended. It is expected that the "off state" will precede the "on state".

4962 **Event Control (mandatory):** Identifies additional control options for the event. The BitMap for
4963 this field is described in Table D-4.

4964 **Table D-4– Event Control Field BitMap**

Bit	Description
0	1= Randomize Start time, 0=Randomized Start not Applied
1	1= Randomize End time, 0=Randomized End not Applied
2 to 7	Reserved

4965
4966 ***Note:*** *The randomization attribute will be used in combination with two bits to determine if*
4967 *the Event Start and Stop Times are randomized. By default devices will randomize the start*
4968 *and stop of an event. Refer to sub-clause D.2.3.2.2 and sub-clause D.2.3.2.3 for the settings of*
4969 *these values.*

4970 ***D.2.2.3.1.1.2 When Generated***

4971 This command is generated when the ESI wants to control one or more load control devices,
4972 usually as the result of an energy curtailment command from the Smart Energy network.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

4973 ***D.2.2.3.1.1.3 Responses to Load Control Event***

4974 The server receives the cluster-specific commands detailed in sub-clause D.2.3.3.1.

4975 D.2.2.3.2 **Cancel Load Control Event Command**

4976 D.2.2.3.2.1 Payload Format

4977 The *Cancel Load Control Event* command payload shall be formatted as illustrated in
4978 Figure D-3.

Octets	**4**	**2**	**1**	**1**	**4**
Data Type	Unsigned 32-bit integer	16-bit BitMap	Unsigned 8-bit integer	8-bit BitMap	UTCTime
Field Name	Issuer Event ID	Device Class (M)	Utility Enrollment Group (M)	Cancel Control (M)	Effective Time (M)

4979 **Figure D-3– *Cancel Load Control Event* Payload**

4980 ***D.2.2.3.2.1.1 Payload Details***

4981 **Issuer Event ID (mandatory):** Unique identifier generated by the Energy provider. The value of
4982 this field allows matching of Event reports with a specific Demand Response and Load Control
4983 event. It's expected the value contained in this field is a unique number managed by upstream
4984 systems or a UTC based time stamp (UTCTime data type) identifying when the Load Control
4985 Event was issued.

4986 **Device Class (mandatory):** Bit encoded field representing the Device Class to apply the
4987 current Load Control Event. Each bit, if set individually or in combination, indicates the class
4988 device(s) needing to participate in the event. (Note that the participating device may be
4989 different than the controlling device. For instance, a thermostat may act on behalf of an HVAC
4990 compressor or furnace and/or Strip Heat/Baseboard Heater and should take action on their
4991 behalf, as the thermostat itself is not subject to load shed but controls devices that are subject to
4992 load shed.) The encoding of the Device Class is listed in Figure D-2.

4993 **Utility Enrollment Group (mandatory):** The Utility Enrollment Group field can be used in
4994 conjunction with the Device Class bits. It provides a mechanism to direct Load Control Events
4995 to groups of Devices. Example, by assigning two different groups relating to either Demand
4996 Response programs or geographic areas, Load Control Events can be further directed for a sub-
4997 set of Device Classes (i.e. Device Class Bit 0 and Utility Enrollment Group #1 vs. Device
4998 Class Bit0 and Utility Enrollment Group #2). 0x00 addresses all groups, and values 0x01 to
4999 0xFF address individual groups that match. Please refer to sub-clause D.2.3.2.1 for further
5000 details.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

If the Device Class and/or Utility Enrollment Group fields don't apply to your End Device, the *Cancel Load Control Event* command is ignored.

Device Class and/or Utility Group fields must be the same for a *Cancel Load Control Event* command as they were for the command to create the event. Should these fields be different there is no defined behavior for how DRLC servers should maintain their tables for replying to *Get Scheduled Events* commands.

Cancel Control (mandatory): The encoding of the Cancel Control is listed in Table D-5.

Table D-5– Cancel Control

Bit	Description
0	To be used when the Event is currently in process and acted upon as specified by the Effective Time field of the *Cancel Load Control Event* command. A value of Zero (0) indicates that randomization is overridden and the event should be terminated immediately at the Effective Time. A value of One (1) indicates the event should end using randomization settings in the original event.
1 to 7	Reserved

Effective Time (mandatory): UTC Timestamp representing when the canceling of the event is scheduled to start. An effective time of 0x00000000 is a special time denoting "now." If the device would send an event with an Effective Time of now, adjust the Duration In Minutes field to correspond to the remainder of the event.
Note: This field is deprecated; a *Cancel Load Control* command shall now take immediate effect. A value of 0x00000000 shall be used in all *Cancel Load Control* commands

D.2.2.3.2.1.2 When Generated

This command is generated when the ESI wants to cancel previously scheduled control of one or more load control devices, usually as the result of an energy curtailment command from the Smart Energy network.

D.2.2.3.2.1.3 Responses to Cancel Load Control Event

The server receives the cluster-specific commands detailed in sub-clause D.2.3.3.1.

Note: *If the Cancel Load Control Event command is received after the event has ended, the device shall reply using the "Report Event Status Command" with an Event Status of "Rejected - Invalid Cancel Command (Undefined Event)".*

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

5025 D.2.2.3.3 **Cancel All Load Control Events Command**

5026 D.2.2.3.3.1 Payload Format

5027 The *Cancel All Load Control Events* command payload shall be formatted as illustrated in
5028 Figure D-4.

5029

Octets	1
Data Type	8-bit BitMap
Field Name	Cancel Control

5030 **Figure D-4– *Cancel All Load Control Events* Command Payload**

5031 ***D.2.2.3.3.1.1 Payload Details***

5032 **Cancel Control:** The encoding of the Cancel Control is listed in Table D-6.

5033 **Table D-6– Cancel All Command Cancel Control Field**

Bit	Description
0	To be used when the Event is currently in process and a cancel command is received. A value of Zero (0) indicates that randomization is overridden and the event should be terminated immediately. A value of One (1) indicates the event should end using randomization settings in the original event.
1 to 7	Reserved

5034

5035 D.2.2.3.3.2 When Generated

5036 This command is generated when the ESI wants to cancel all events for control device(s).

5037 D.2.2.3.3.3 Responses to Cancel All Load Control Events

5038 The server receives the cluster-specific commands detailed in sub-clause D.2.3.3.1. The *Cancel*
5039 *All Load Control Events* command is processed by the device as if individual *Cancel Load*
5040 *Control Event* commands were received for all of the currently stored events in the device. The
5041 device will respond with a “Report Event Status Command” for each individual load control
5042 event canceled.

5043 D.2.2.4 Commands Received

5044 The server receives the cluster-specific commands detailed in sub-clause D.2.3.3.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

D.2.3 Client

This section identifies the attributes and commands provided by Client devices.

D.2.3.1 Dependencies

Devices receiving and acting upon *Load Control Event* commands must be capable of storing and supporting at least three unique instances of events. As a highly recommended recovery mechanism, when maximum storage of events has been reached and additional Load Control Events are received that are unique (not superseding currently stored events), devices should ignore additional Load Control Events and when storage becomes available, utilize the *GetScheduledEvents* command to retrieve any previously ignored events.

Events carried using this cluster include a timestamp with the assumption that target devices maintain a real time clock. Devices can acquire and synchronize their internal clocks with the ESI as described in sub-clause 5.12.1.1.

Devices MAY 'drop' events received before they have received and resolved time ('dropping' an event is defined as sending a default response with status code SUCCESS).

If a device does not support a real time clock, it's assumed the device will ignore all values within the Time field except the "Start Now" value.

Additionally, for devices without a real time clock it's assumed those devices will utilize a method (i.e. ticks, countdowns, etc.) to approximate the correct duration period.

D.2.3.2 Client Cluster Attributes

Table D-7– Demand Response Client Cluster Attributes

Identifier	Name	Type	Range	Access	Default	Mandatory/ Optional
0x0000	*UtilityEnrollment Group*	Unsigned 8-bit Integer	0x00 to 0xFF	Read/ Write	0x00	M
0x0001	*StartRandomizeMinutes*	Unsigned 8-bit Integer	0x00 to 0x3C	Read/ Write	0x1E	M
0x0002	*StopRandomizeM nutes*	Unsigned 8-bit Integer	0x00 to 0x3C	Read/ Write	0x1E	M
0x0003	*DeviceClassValue*	Unsigned 16-bit Integer	0x0000 to 0xFFFF	Read/ Write	-	M
0x0004 to 0xFFFF	Reserved					

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

5066 D.2.3.2.1 **Utility Enrollment Group Attribute**

5067 The *UtilityEnrollmentGroup* provides a method for utilities to assign devices to groups. In
5068 other words, Utility defined groups provide a mechanism to arbitrarily group together different
5069 sets of load control or demand response devices for use as part of a larger utility program. The
5070 definition of the groups, implied usage, and their assigned values are dictated by the Utilities
5071 and subsequently used at their discretion, therefore outside the scope of this specification. The
5072 valid range for this attribute is 0x00 to 0xFF, where 0x00 (the default value) indicates the device
5073 is a member of all groups and values 0x01 to 0xFF indicates that the device is member of that
5074 specified group.

5075 D.2.3.2.2 **Start Randomization Minutes Attribute**

5076 The *StartRandomizedMinutes* represents the maximum number of minutes to be used when
5077 randomizing the start of an event. As an example, if *StartRandomizedMinutes* is set for 3
5078 minutes, the device could randomly select 2 minutes (but never greater than the 3 minutes) for
5079 this event, causing the start of the event to be delayed by two minutes. The valid range for this
5080 attribute is 0x00 to 0x3C where 0x00 indicates start event randomization is not performed.

5081 D.2.3.2.3 **End Randomization Minutes Attribute**

5082 The *EndRandomizedMinutes* represents the maximum number of minutes to be used when
5083 randomizing the end of an event. As an example, if *EndRandomizedMinutes* is set for 3 minutes,
5084 the device could randomly select one minute (but never greater than 3 minutes) for this event,
5085 causing the end of the event to be delayed by one minute. The valid range for this attribute is
5086 0x00 to 0x3C where 0x00 indicates end event randomization is not performed.

5087 D.2.3.2.4 **DeviceClassValue Attribute**

5088 The *DeviceClassValue* attribute identifies which bits the device will match in the Device Class
5089 fields. Please refer to Table D-2, "Device Class Field BitMap/ Encoding" for further details.
5090 Although the attribute has a read/write access property, the device is permitted to refuse to
5091 change the *DeviceClass* by setting the status field of the corresponding write attribute
5092 status record to NOT_AUTHORIZED.

5093 Although, for backwards compatibility, the Type cannot be changed, this 16-bit Integer should be
5094 treated as if it were a 16-bit BitMap.

5095 Device Class and/or Utility Enrollment Group fields are to be used as filters for deciding to
5096 accept or ignore a *Load Control Event* or a *Cancel Load Control Event* command. There is no
5097 requirement for a device to store or remember the Device Class and/or Utility Enrollment Group
5098 once the decision to accept the event has been made. A consequence of this is that devices that
5099 accept multiple device classes may have an event created for one device class superseded by an
5100 event created for another device class.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

In-Home Displays should report the device classes that they are interested in. An IHD that wishes to display all possible Load Control Events, even for classes not yet defined, should indicate a device class of 0xFFFF; this will allow DRLC servers to optimize the number of DRLC events they unicast, such that they are only sent to those devices that are interested in them.

D.2.3.3 Commands Generated

The command IDs generated by the Demand Response and Load Control client cluster are listed in Table D-8.

Table D-8– Generated Command IDs for the Demand Response and Load Control Client

Command Identifier Field Value	Description	Mandatory/ Optional
0x00	*Report Event Status*	M
0x01	*Get Scheduled Events*	M
0x02 – 0xff	Reserved	

D.2.3.3.1 **Report Event Status Command**

D.2.3.3.1.1 Payload Format

The *Report Event Status* command payload shall be formatted as illustrated in Figure D-5.

Octets	**4**	**1**	**4**	**1**	**2**	**2**
Data Type	Unsigned 32-bit integer	Unsigned 8-bit integer	UTCTime	Unsigned 8-bit integer	Unsigned 16-bit integer	Unsigned 16-bit integer
Field Name	Issuer Event ID (M)	Event Status (M)	Event Status Time (M)	Criticality Level Applied (M)	Cooling Temperature Set Point Applied (O)	Heating Temperature Set Point Applied (O)

Octets	**1**	**1**	**1**	**1**	**42**
Data Type	Signed 8-bit integer	Unsigned 8-bit integer	8-bit BitMap	Unsigned 8-bit integer	Octets (non- ZCL Data Type)

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Field Name	Average Load Adjustment Percentage Applied (O)	Duty Cycle Applied (O)	Event Control (M)	Signature Type (M)	Signature (O)

Figure D-5– *Report Event Status* Command Payload

D.2.3.3.1.1.1 Payload Details

Issuer Event ID (mandatory): Unique identifier generated by the Energy provider. The value of this field allows matching of Event reports with a specific Demand Response and Load Control event. It's expected the value contained in this field is a unique number managed by upstream systems or a UTC based time stamp (UTCTime data type) identifying when the Load Control Event was issued.

Event Status (mandatory): Table D-9 lists the valid values returned in the Event Status field.

Table D-9– Event Status Field Values

Value	Description
0x00	Reserved for future use.
0x01	*Load Control Event* command received
0x02	Event started
0x03	Event completed
0x04	User has chosen to “Opt-Out”, user will not participate in this event
0x05	User has chosen to “Opt-In”, user will participate in this event
0x06	The event has been cancelled
0x07	The event has been superseded
0x08	Event partially completed with User “Opt-Out”.
0x09	Event partially completed due to User “Opt-In”.
0x0A	Event completed, no User participation (Previous “Opt-Out”).
0x0B to 0xF7	Reserved for future use.
0xF8	Rejected - Invalid Cancel Command (Default)
0xF9	Rejected - Invalid Cancel Command (Invalid Effective Time)
0xFA	Reserved
0xFB	Rejected - Event was received after it had expired (Current Time > Start Time + Duration)
0xFC	Reserved for future use.
0xFD	Rejected - Invalid Cancel Command (Undefined Event)
0xFE	*Load Control Event* command Rejected
0xFF	Reserved for future use.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Should a device issue one or more "OptOut" or "OptIn" RES commands during an event that is eventually cancelled, the event shall be recorded as a cancelled event (Status = 0x06) at its effective time.

Should a device issue one or more "OptOut" or "OptIn" RES commands during an event that is not cancelled, the event shall be recorded as partially completed based on the last RES command sent (Status = 0x08 or 0x09).

When a device returns a status of 0xFD (Rejected - Invalid Cancel Command (Undefined Event)), all optional fields should report their "Ignore" values.

When a device receives a duplicate RES command, it should ignore the duplicate commands. Please note: As a recommended best practice, ESI applications should provide a mechanism to assist in filtering duplicate messages received on the WAN.

Event Status Time (mandatory): UTC Timestamp representing when the event status occurred. This field shall not use the value of 0x00000000.

Criticality Level Applied (mandatory): Criticality Level value applied by the device, see the corresponding field in the *Load Control Event* command for more information.

Cooling Temperature Set Point Applied (optional): Cooling Temperature Set Point value applied by the device, see the corresponding field in the *Load Control Event* command for more information. The value 0x8000 means that this field has not been used by the end device.

Heating Temperature Set Point Applied (optional): Heating Temperature Set Point value applied by the device, see the corresponding field in the *Load Control Event* command for more information. The value 0x8000 means that this field has not been used by the end device.

Average Load Adjustment Percentage Applied (optional): Average Load Adjustment Percentage value applied by the device, see the corresponding field in the *Load Control Event* command for more information. The value 0x80 means that this field has not been used by the end device.

Duty Cycle Applied (optional): Defines the maximum On state duty cycle applied by the device. The value 0xFF means that this field has not been used by the end device. Refer to sub-clause D.2.2.3.1.1.1.

Event Control (mandatory): Identifies additional control options for the event. Refer to sub-clause D.2.2.3.1.1.1.

Signature Type (mandatory): An 8-bit Unsigned integer enumerating the type of algorithm use to create the Signature. The enumerated values are:

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Enumerated Value	Signature Type
0x00	No Signature
0x01	ECDSA
0x02 to 0xFF	Reserved

If the signature field is not used, the signature type shall be set to 0x00, which will be used to indicate "no signature". The signature field shall be filled with (48) 0xFF values.

Signature (optional): A non-repudiation signature created by using the Matyas-Meyer-Oseas hash function (specified in Annex B.6 in [B3]) used in conjunction with ECDSA. The signature creation process will occur in two steps:

1 Pass the first ten fields, which includes all fields up to the Signature field, of the *Report Event Status* command (listed in Figure D-5) through ECDSA using the device's ECC Private Key, generating the signature *(r,s)*.
Note: *ECDSA internally uses the MMO hash function in place of the internal SHA-1 hash function.*

2 Concatenate ECDSA signature components *(r,s)* and place into the Signature field within the *Report Event Status* command.
Note: *the lengths of r and s are implicit, based on the curve used. Verifying the signature will require breaking the signature field back into the discrete components r and s, based on the length.*

D.2.3.3.1.2 When Generated

This command is generated when the client device detects a change of state for an active Load Control event. (The transmission of this command should be delayed after a random delay between 0 and 5 seconds, to avoid a potential storm of packets.)

D.2.3.3.2 **Get Scheduled Events Command**

Note: *The handling of this command is currently under review, and is likely to change in the next revision of the specification. Refer to CCB 1297 (and associated document 12-0180-00) for further information*

This command is used to request that all scheduled Load Control Events, starting at or after the supplied Start Time, are re-issued to the requesting device. When received by the Server, one or more *Load Control Event* commands (see sub-clause D.2.2.3.1) will be sent covering both active and scheduled Load Control Events.

D.2.3.3.2.1 Payload Format

The *Get Scheduled Events* command payload shall be formatted as illustrated in Figure D-6

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Octets	4	1
Data Type	UTCTime	Unsigned 8-bit integer
Field Name	Start Time (M)	Number of Events (M)

Figure D-6– *Get Scheduled Events* Command Payload

Start Time (mandatory): UTC Timestamp representing the minimum ending time for any scheduled or currently active events to be resent. If either command has a Start Time of 0x00000000, replace that Start Time with the current time stamp.

Number of Events (mandatory): Represents the maximum number of events to be sent. A value of 0 would indicate all available events are to be returned. Example: Number of Events = 1 would return the first event with an EndTime greater than or equal to the value of Start Time field in the *Get Scheduled Events* command (EndTime would be StartTime plus Duration of the event listed in the device's event table).

D.2.3.3.2.2 When Generated

This command is generated when the client device wishes to verify the available Load Control Events or after a loss of power/reset occurs and the client device needs to recover currently active or scheduled Load Control Events.

A ZCL Default Response with status NOT_FOUND shall be returned when there are no events available.

D.2.3.4 Commands Received

The client receives the cluster-specific commands detailed in sub-clause D.2.2.

D.2.3.5 Attribute Reporting

Attribute reporting is not expected to be used for this cluster. The Client side attributes are not expected to be changed by the Client, only used during Client operations.

D.2.4 Application Guidelines

The criticality level is sent by the utility to the load control device to indicate how much load reduction is requested. The utility is not required to use all of the criticality levels that are described in this specification. A load control device is not required to provide a unique response to each criticality level that it may receive.

The Average Load Adjustment Percentage, temperature offsets, and temperature set points are used by load control devices and energy management systems on a “voluntary” or “optional” basis. These devices are not required to use the values that are provided by the utility. They are provided as a recommendation by the utility.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

The load control device shall, in a manner that is consistent with this specification, accurately report event participation by way of the Report Event Status message.

The Average Load Adjustment Percentage is sent by the utility to the load control device to indicate how much load reduction is requested. The load control device may respond to this information in a unique manner as defined by the device manufacturer.

The Duty Cycle is sent by the utility to the load control device to indicate the maximum "On state" for a device. The control device may respond to this information in a unique manner as defined by the device manufacturer.

The cooling temperature offset may be sent by the utility to the load shed control to indicate how much indoor cooling temperature offset is requested. Response of a load control device to this information is not mandatory. The control device may respond to this information in a unique manner as defined by the device manufacturer.

The heating temperature offset may be sent by the utility to the load control device to indicate how much indoor heating temperature offset is requested. The control device may respond to this information in a unique manner as defined by the device manufacturer.

The cooling temperature may be sent by the utility to the load control device to indicate the indoor cooling temperature setting that is requested. The control device may respond to this information in a unique manner as defined by the device manufacturer.

The heating temperature may be sent by the utility to the load control device to indicate the indoor heating temperature setting that is requested. The control device may respond to this information in a unique manner as defined by the device manufacturer.

Note: *The most recent Load Control Event supersedes any previous Load Control Event command for the set of Device Classes and groups for a given time. Nested events and overlapping events are not allowed. The current active event will be terminated if a new event is started.*

D.2.4.1 Load Control Rules, Server

D.2.4.1.1 **Load Control Server, Identifying Use of SetPoint and Offset Fields**

The use of the fields, Heating and Cooling Temperature Set Points and Heating and Cooling Temperature Offsets is optional. All fields in the payload must be populated. Non-use of these fields by the Server is indicated by using the following values: 0x8000 for Set Points and 0xFF for Offsets. When any of these four fields are indicated as optional, they shall be ignored by the client.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

5249 D.2.4.1.2 **Load Control Server, Editing of Scheduled Events**

5250 Editing of a scheduled demand response event is not allowed. Editing of an active demand
5251 response event is not allowed. Nested events and overlapping events are not allowed. The
5252 current active event will be terminated if a new event is started.

5253 ### D.2.4.2 Load Control Rules, Client

5254 D.2.4.2.1 **Start and Stop Randomization**

5255 When shedding loads (turning a load control device off), the load control device will optionally
5256 apply start time randomization based on the values specified in the Event Control Bits and the
5257 Client's *Start Randomization Minutes* attribute. By default, devices will apply a random delay
5258 as specified by the default values of start and end randomization in the Demand Response
5259 Client Cluster Attributes table.

5260 When ending a load control event, the load control device will support the same randomization
5261 features as provided in the start load control event.

5262 D.2.4.2.2 **Editing of DR Control Parameters**

5263 In Load Control Device and energy management systems, editing of the demand response
5264 control parameters while participating in an active demand response event is not allowed.

5265 D.2.4.2.3 **Response to Price Events + Load Control Events**

5266 The residential system's response to price driven events will be considered in addition to the
5267 residential system's response to demand response events. Demand response events which require
5268 that the residential system is turned off have priority over price driven events. Demand response
5269 events which require that the residential system go to a fixed setting point have priority over price
5270 driven events. In this case, the thermostat shall not use a Cooling or Heating Temperature Set
5271 Point that causes the device to use more energy than the price driven event setting.

5272 D.2.4.2.4 **Opt-Out Messages**

5273 An event override message, "opt-out", will be sent by the load control device or energy
5274 management system if the operator chooses not to participate in a demand response event by
5275 taking action to override the programmed demand reduction response. The override message
5276 will be sent at the start of the event. In the case where the event has been acknowledged and
5277 started, the override message will be sent when the override occurs.

5278 D.2.4.2.5 **Thermostat/HVAC Controls**

5279 A residential HVAC system will be allowed to change mode, from off to Heat, off to Cool, Cool
5280 to Heat, or Heat to Cool, during a voluntary event which is currently active. The HVAC control
5281 must acknowledge the event, as if it was operating, in that mode, at the start of the event. The

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

HVAC control must obey the event rules that would have been enforced if the system had been operating in that mode at the start of the active event.

An event override message, “opt-out”, will be sent by the load control device or energy management system if the operator chooses not to participate in a demand response event by taking action to override the programmed demand reduction response. The override message will be sent at the start of the event. In the case where the event has been acknowledged and started, the override message will be sent when the override occurs.

D.2.4.2.6 Demand Response and Load Control Transaction Examples

The following example in Figure D-7 depicts the transactions that would take place for two events, one that is successful and another that is overridden by the user.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

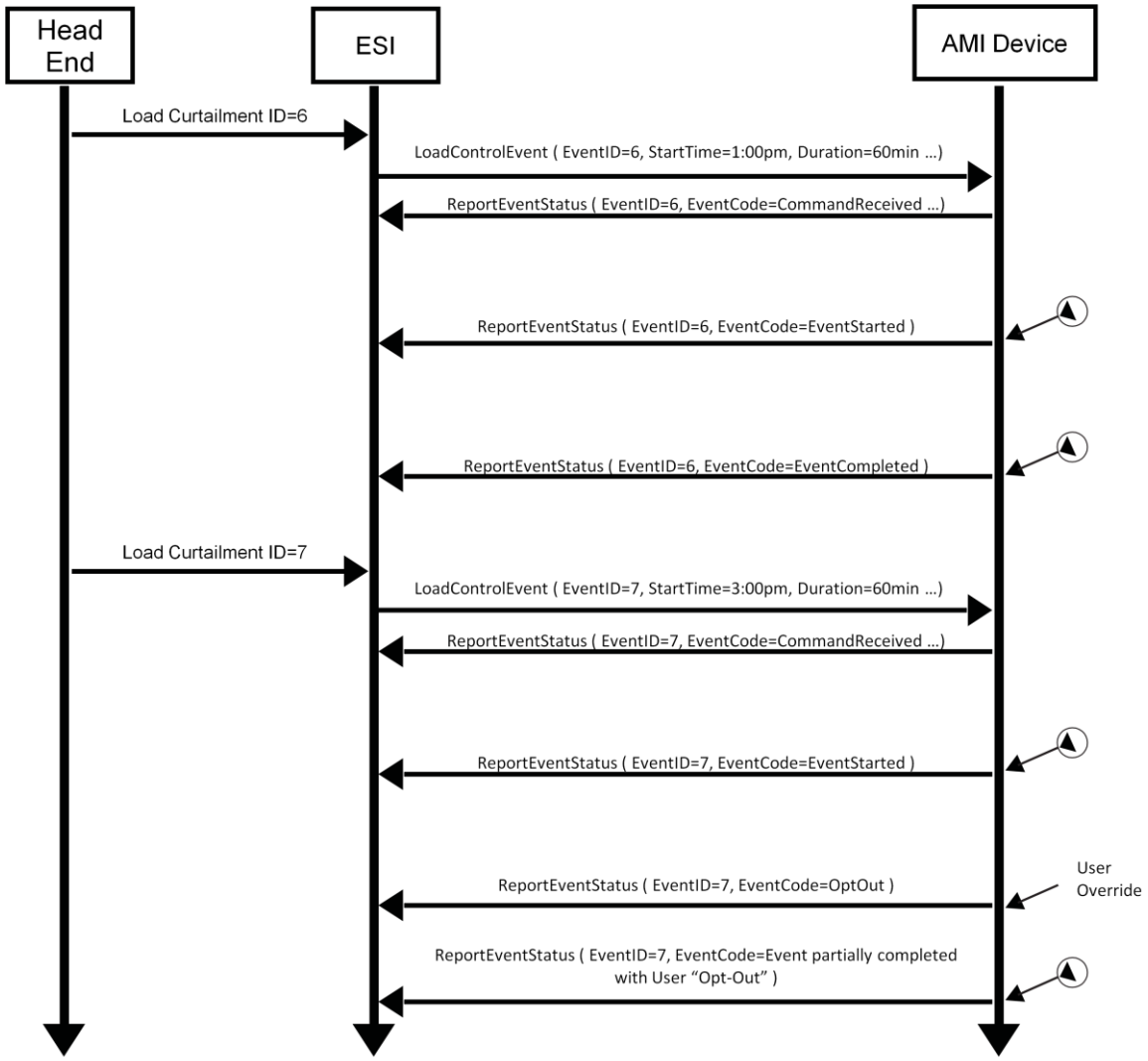


Figure D-7– Example of Both a Successful and an Overridden Load Curtailment Event

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

The example in Figure D-8 depicts the transactions that would take place when an event is superseded by an event that is eventually cancelled.

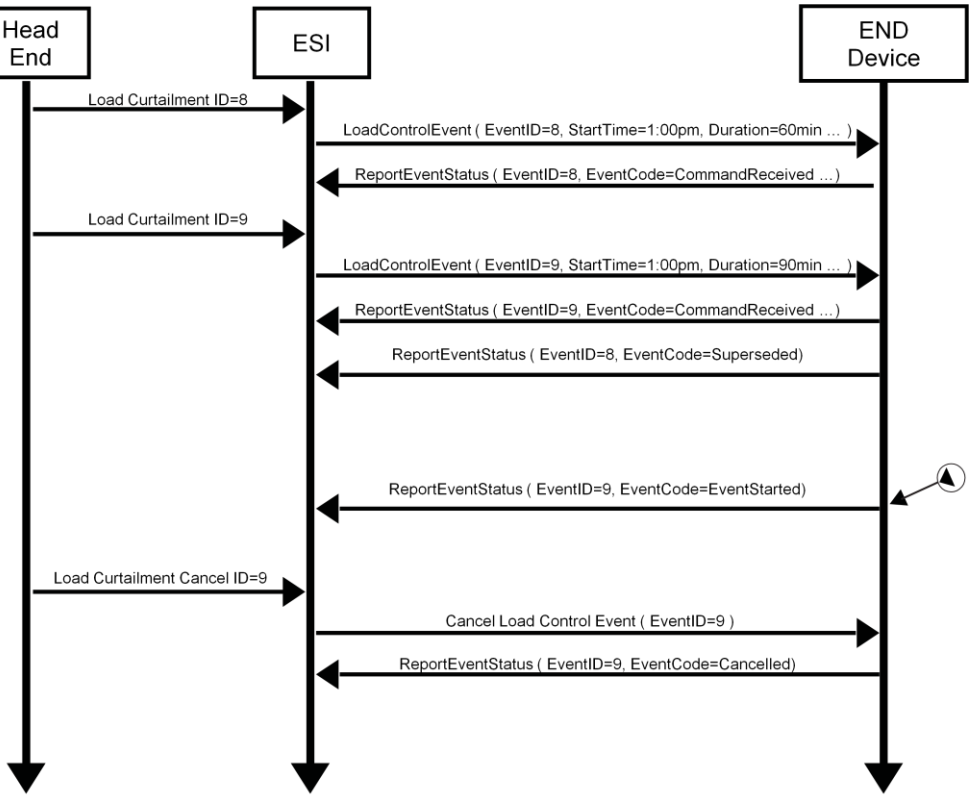


Figure D-8– Example of a Load Curtailment Superseded and Another Cancelled

Please refer to Annex E for more information regarding the management and behavior of overlapping events.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

D.3 Metering Cluster

D.3.1 Overview

The Metering Cluster provides a mechanism to retrieve usage information from Electric, Gas, Water, and potentially Thermal metering devices. These devices can operate on either battery or mains power, and can have a wide variety of sophistication. The Metering Cluster is designed to provide flexibility while limiting capabilities to a set number of metered information types. More advanced forms or data sets from metering devices will be supported in the Smart Energy Tunneling Cluster, which will be defined in sub-clause D.6.

The following figures identify three configurations as examples utilizing the Metering Cluster.

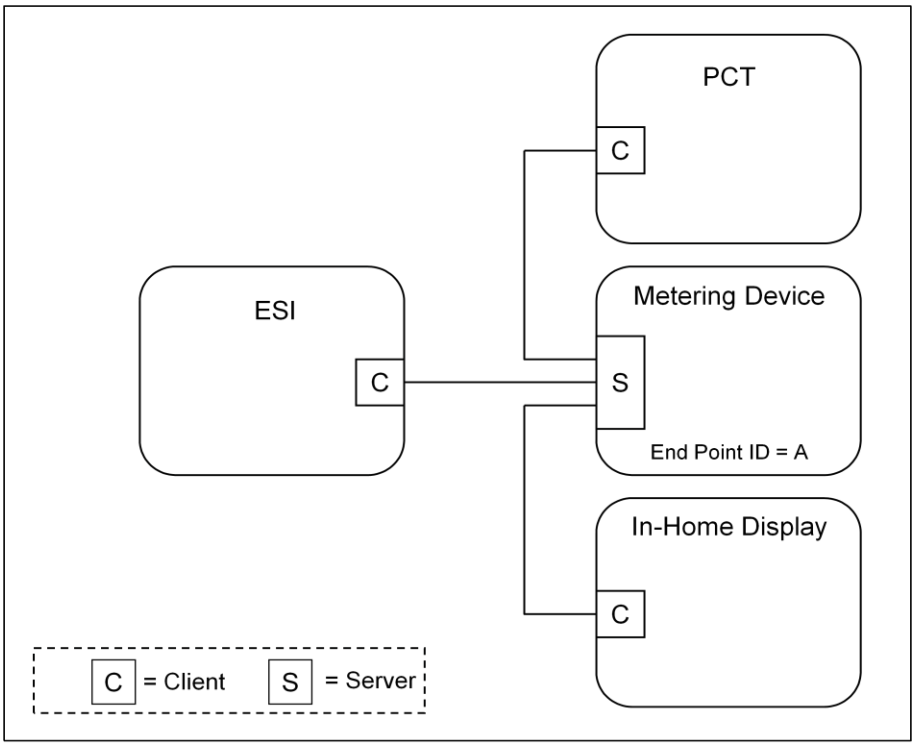


Figure D-9– Standalone ESI Model with Mains Powered Metering Device

In the example shown in Figure D-9, the metering device is the source of information provided via the Metering Cluster Server.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

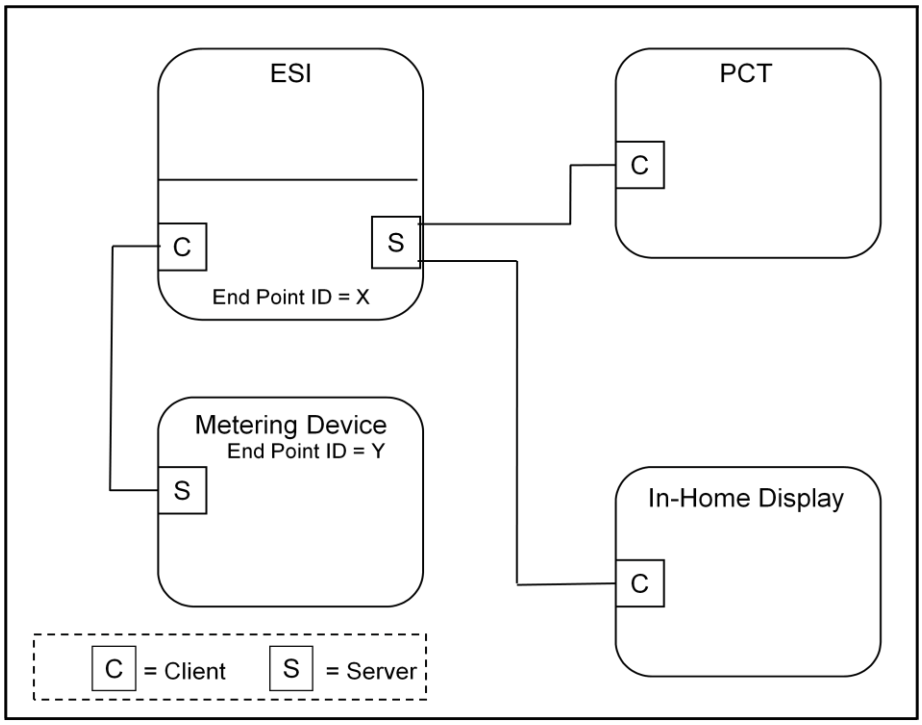


Figure D-10– Standalone ESI Model with Battery Powered Metering Device

In the example shown in Figure D-10, the metering device is running on battery power and its duty cycle for providing information is unknown. It's expected the ESI will act like a mirrored image or a mailbox (Client) for the metering device data, allowing other Smart Energy devices to gain access to the metering device's data (provided via an image of its Metering Cluster).

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

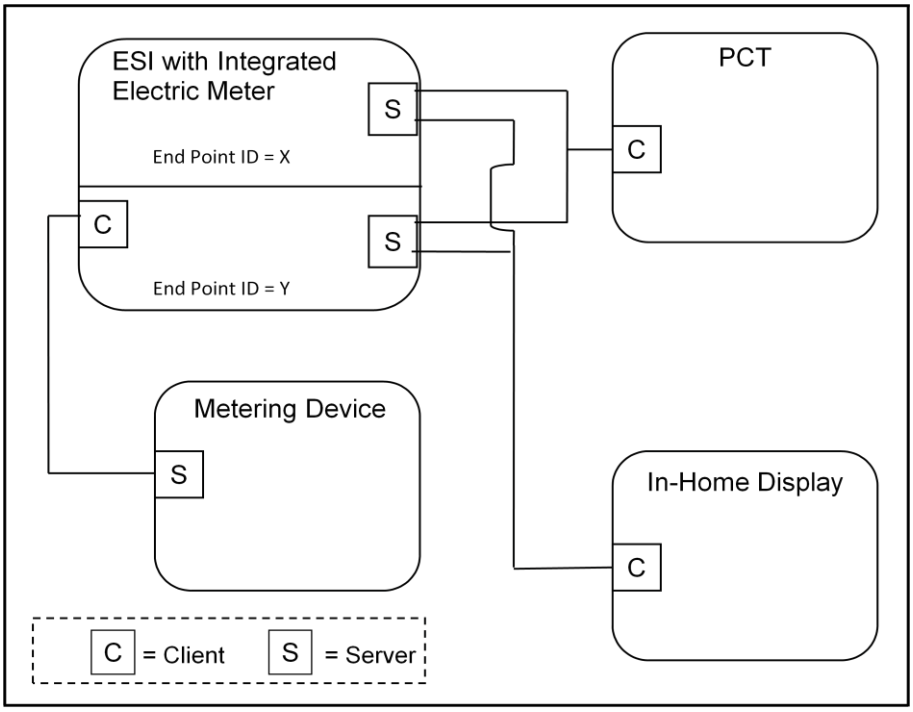


Figure D-11– ESI Model with Integrated Metering Device

In the example shown in Figure D-11, much like the previous example in Figure D-10, the external metering device is running on battery power and its duty cycle for providing information is unknown. It's expected the ESI will act like a Client side mailbox for the external metering device data, allowing other Smart Energy devices to gain access to the metering device's data (provided via an image of its Metering Cluster). Since the ESI can also contain an integrated metering device where its information is also conveyed through the Metering Cluster, each device (external metering device mailbox and integrated meter) will be available via independent EndPoint IDs. Other Smart Energy devices that need to access the information must understand the ESI cluster support by performing service discoveries. It can also identify if an Endpoint ID is a mailbox/ mirror of a metering device by reading the *MeteringDeviceType* attribute (refer to sub-clause D.3.2.2.4.7).

In the above examples (Figure D-10 and Figure D-11), it is expected the ESI would perform Attribute Reads (or configure Attribute Reporting) and use the *GetProfile* command to receive the latest information whenever the Metering Device (EndPoint Z) wakes up. When received, the ESI will update its mailbox (EndPoint ID Y in Figure D-10 and Figure D-11) to reflect the latest data available. A metering device using the mirror is also allowed (and recommended) to push metering data updates to the ESI via *Report Attribute* commands as described in sub-clause D.3.4.4.

Other Smart Energy devices can access EndPoint Y in the ESI to receive the latest information just as they would to access information in the ESI's integrated Electric meter (as in Figure D-11, EndPoint X) and other Metering devices (as in Figure D-9, EndPoint A).

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

D.3.2 Server

D.3.2.1 Dependencies

Subscribed reporting of Metering attributes.

D.3.2.2 Attributes

For convenience, the attributes defined in this specification are arranged into sets of related attributes; each set can contain up to 256 attributes. Attribute identifiers are encoded such that the most significant Octet specifies the attribute set and the least significant Octet specifies the attribute within the set. The currently defined attribute sets are listed in Table D-10.

Note: *Certain attributes within this cluster are provisionary and not certifiable. Refer to the individual attribute sets for details of the relevant attributes.*

Table D-10– Metering Cluster Server Attribute Sets

Attribute Set Identifier	Description
0x00	Reading Information Set
0x01	TOU Information Set
0x02	Meter Status
0x03	Formatting
0x04	Historical Consumption
0x05	Load Profile Configuration
0x06	Supply Limit
0x07	Block Information (Delivered)
0x08	Alarms
0x09	Block Information (Received)
0x0A	Meter Billing Attribute Set
0x0B	Supply Control Attribute Set
0x0C	Alternative Historical Consumption
0x0D to 0xFF	Reserved

D.3.2.2.1 Reading Information Set

The following set of attributes provides a remote access to the reading of the Electric, Gas, or Water metering device. A reading must support at least one register which is the actual total summation of the delivered quantity (kWh, m^3, ft^3, ccf, US gl).

Please note: In the following attributes, the term “Delivered” refers to the quantity of Energy, Gas, or Water that was delivered to the customer from the utility. Likewise, the term

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

5379 "Received" refers to the quantity of Energy, Gas, or Water that was received by the utility from
5380 the customer.

5381 ***Note:*** *Metering Cluster Reading Attribute 0x0012 in this revision of this specification is*
5382 *provisionary and not certifiable. This feature set may change before reaching certifiable status*
5383 *in a future revision of this specification.*

5384 **Table D-11– Reading Information Attribute Set**

Identifier	Name	Type	Range	Access	Default	Man. /Opt.
0x0000	*CurrentSummation Delivered*	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	M
0x0001	*CurrentSummation Received*	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0002	*CurrentMaxDemand Delivered*	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0003	*CurrentMaxDemandReceived*	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0004	*DFTSummation*	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0005	*Daily Freeze Time*	Unsigned 16-bit Integer	0x0000 to 0x173B	Read Only	0x0000	O
0x0006	*PowerFactor*	Signed 8-bit Integer	-100 to +100	Read Only	0x00	O
0x0007	*ReadingSnapshot Time*	UTCTime		Read Only	-	O
0x0008	*CurrentMaxDemand DeliveredTime*	UTCTime		Read Only	-	O
0x0009	*CurrentMaxDemandReceivedTime*	UTCTime		Read Only	-	O
0x000A	*DefaultUpdate Period*	Unsigned 8-bit Integer	0x00 to 0xFF	Read Only	0x1E	O
0x000B	*FastPollUpdate Period*	Unsigned 8-bit Integer	0x00 to 0xFF	Read Only	0x05	O
0x000C	*CurrentBlockPeriod ConsumptionDelivered*	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x000D	*DailyConsumption Target*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x000E	*CurrentBlock*	8-bit Enumeration	0x00 to 0x10	Read Only	-	O
0x000F	*ProfileInterval Period*	8-bit Enumeration	0x00 to 0xFF	Read Only	-	O
0x0010	*Deprecated*[3]					
0x0011	*PresetReading Time*	Unsigned 16-bit Integer	0x0000 to 0x173B	Read Only	0x0000	O
0x0012	*VolumePerReport*	Unsigned 16-bit Integer	0x0000 to 0xFFFF	Read Only	-	O
0x0013	*FlowRestriction*	Unsigned 8-bit Integer	0x00 to 0xFF	Read Only	-	O
0x0014	*Supply Status*	8-bit Enumeration	0x00 to 0xFF	Read Only	-	O
0x0015	*CurrentInletEnergy CarrierSummation*	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O[4]
0x0016	*CurrentOutletEnergy Carrier Summation*	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0017	*InletTemperature*	Signed 24-bit Integer	-8,388,607 to 8,388,607	Read Only	-	O[5]
0x0018	*OutletTempera ture*	Signed 24-bit Integer	-8,388,607 to 8,388,607	Read Only	-	O[6]
0x0019	*ControlTemperature*	Signed 24-bit Integer	-8,388,607 to 8,388,607	Read Only	-	O

[3] CCB 1886
[4] CCB 1999
[5] CCB 1999
[6] CCB 1999

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x001A	*CurrentInletEnergy CarrierDemand*	Signed 24-bit Integer	-8,388,607 to 8,388,607	Read Only	-	O
0x001B	*CurrentOutletEnergy Carrier Demand*	Signed 24-bit Integer	-8,388,607 to 8,388,607	Read Only	-	O
0x001C	*PreviousBlockPeriod ConsumptionDelivered*	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x001D	CurrentBlockPeriod ConsumptionReceived	Unsigned 48 bit integer	0x000000000000 – 0xFFFFFFFFFFFF	Read only	-	O
0x001E	CurrentBlockReceived	8-bit Enumeration	0x00 – 0xFF	Read Only	-	O
0x001F	DFTSummation Received	Unsigned 48 bit integer	0x000000000000 – 0xFFFFFFFFFFFF	Read Only	-	O
0x0020	ActiveRegisterTier Delivered	8-bit Enumeration	0 – 48	Read Only	-	O
0x0021	ActiveRegisterTier Received	8-bit Enumeration	0 – 48	Read Only	-	O
0x0022	LastBlockSwitchTime	UTCTime		Read Only	-	O
0x0023 to 0x00FF	Reserved					

D.3.2.2.1.1 CurrentSummationDelivered Attribute

CurrentSummationDelivered represents the most recent summed value of Energy, Gas, or Water delivered and consumed in the premises. *CurrentSummationDelivered* is mandatory and must be provided as part of the minimum data set to be provided by the metering device. *CurrentSummationDelivered* is updated continuously as new measurements are made.

D.3.2.2.1.2 CurrentSummationReceived Attribute

CurrentSummationReceived represents the most recent summed value of Energy, Gas, or Water generated and delivered from the premises. If optionally provided, *CurrentSummationReceived* is updated continuously as new measurements are made.

D.3.2.2.1.3 CurrentMaxDemandDelivered Attribute

CurrentMaxDemandDelivered represents the maximum demand or rate of delivered value of Energy, Gas, or Water being utilized at the premises. If optionally provided, *CurrentMaxDemandDelivered* is updated continuously as new measurements are made.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

5399 D.3.2.2.1.4 CurrentMaxDemandReceived Attribute

5400 *CurrentMaxDemandReceived* represents the maximum demand or rate of received value of
5401 Energy, Gas, or Water being utilized by the utility. If optionally provided,
5402 *CurrentMaxDemandReceived* is updated continuously as new measurements are made.

5403 D.3.2.2.1.5 DFTSummation Attribute

5404 *DFTSummation* represents a snapshot of attribute *CurrentSummationDelivered* captured at the
5405 time indicated by attribute *DailyFreezeTime*. If optionally provided, *DFTSummation* is updated
5406 once every 24 hours and captured at the time set in sub-clause D.3.2.2.1.6.

5407 D.3.2.2.1.6 DailyFreezeTime Attribute

5408 *DailyFreezeTime* represents the time of day when *DFTSummation* is captured. *DailyFreezeTime*
5409 is an unsigned 16-bit value representing the hour and minutes for DFT. The byte usages are:

5410 **Bits 0 to 7:** Range of 0 to 0x3B representing the number of minutes past the top of the hour.
5411
5412 **Bits 8 to 15:** Range of 0 to 0x17 representing the hour of the day (in 24-hour format). *Note*
5413 *that midnight shall be represented as 00:00 only*.

5414 D.3.2.2.1.7 PowerFactor Attribute

5415 *PowerFactor* contains the Average Power Factor ratio in 1/100ths. Valid values are 0 to 99.

5416 D.3.2.2.1.8 ReadingSnapshotTime Attribute

5417 The *ReadingSnapshotTime* attribute represents the last time all of the
5418 *CurrentSummationDelivered*, *CurrentSummationReceived*, *CurrentMaxDemandDelivered*, and
5419 *CurrentMaxDemandReceived* attributes that are supported by the device were updated.

5420 D.3.2.2.1.9 CurrentMaxDemandDeliveredTime Attribute

5421 The *CurrentMaxDemandDeliveredTime* attribute represents the time when
5422 *CurrentMaxDemandDelivered* reading was captured.

5423 D.3.2.2.1.10 CurrentMaxDemandReceivedTime Attribute

5424 The *CurrentMaxDemandReceivedTime* attribute represents the time when
5425 *CurrentMaxDemandReceived* reading was captured.

5426 D.3.2.2.1.11 DefaultUpdatePeriod Attribute

5427 The *DefaultUpdatePeriod* attribute represents the interval (seconds) at which the
5428 *InstantaneousDemand* attribute is updated when not in fast poll mode. *InstantaneousDemand*
5429 may be continuously updated as new measurements are acquired, but at a minimum

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

InstantaneousDemand must be updated at the *DefaultUpdatePeriod*. The *DefaultUpdatePeriod* may apply to other attributes as defined by the device manufacturer.

D.3.2.2.1.12 FastPollUpdatePeriod Attribute

The *FastPollUpdatePeriod* attribute represents the interval (seconds) at which the *InstantaneousDemand*attribute is updated when in fast poll mode. *InstantaneousDemand* may be continuously updated as new measurements are acquired, but at a minimum, *InstantaneousDemand* must be updated at the *FastPollUpdatePeriod*. The *FastPollUpdatePeriod* may apply to other attributes as defined by the device manufacturer.

D.3.2.2.1.13 CurrentBlockPeriodConsumptionDelivered Attribute

The *CurrentBlockPeriodConsumptionDelivered* attribute represents the most recent summed value of Energy, Gas or Water delivered and consumed in the premises during the Block Tariff Period.

The *CurrentBlockPeriodConsumptionDelivered* is reset at the start of each Block Tariff Period.

D.3.2.2.1.14 DailyConsumptionTarget Attribute

The *DailyConsumptionTarget* attribute is a daily target consumption amount that can be displayed to the consumer on a HAN device, with the intent that it can be used to compare to actual daily consumption (e.g. compare to the *CurrentDayConsumptionDelivered*).

This may be sent from the utility to the ESI, or it may be derived. Although intended to be based on Block Thresholds, it can be used for other targets not related to blocks. The formatting will be based on the *HistoricalConsumptionFormatting* attribute.

Example: If based on a Block Threshold, the *DailyConsumptionTarget* could be calculated based on the number of days specified in the Block Tariff Period and a given Block Threshold as follows: *DailyConsumptionTarget = BlockNThreshold /* ((*BlockPeriodDuration* /60) / 24). Example: If the target is based on a *Block1Threshold* of 675kWh and where 43200 *BlockThresholdPeriod* is the number of minutes in the billing period (30 days), the *ConsumptionDailyTarget* would be 675 / ((43200 / 60) / 24) = 22.5 kWh per day.

D.3.2.2.1.15 CurrentBlock Attribute

When Block Tariffs are enabled, *CurrentBlock* is an 8-bit Enumeration which indicates the currently active block. If blocks are active then the current active block is based on the *CurrentBlockPeriodConsumptionDelivered* and the block thresholds. Block 1 is active when the value of *CurrentBlockPeriodConsumptionDelivered* is less than or equal to the[7] *Block1Threshold* value, Block 2 is active when *CurrentBlockPeriodConsumptionDelivered* is greater than *Block1Threshold* value and less than or equal to the[8] *Block2Threshold* value, and so

[7] CCB 1679
[8] CCB 1679

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

on. Block 16 is active when the value of *CurrentBlockPeriodConsumptionDelivered* is greater than *Block15Threshold* value.

Table D-12– Block Enumerations

Enumerated Value	Register Block
0x00	No Blocks in use
0x01	Block1
0x02	Block2
0x03	Block3
0x04	Block4
0x05	Block5
0x06	Block6
0x07	Block7
0x08	Block8
0x09	Block9
0x0A	Block10
0x0B	Block11
0x0C	Block12
0x0D	Block13
0x0E	Block14
0x0F	Block15
0x10	Block16
0x11 to 0xFF	Reserved

D.3.2.2.1.16 ProfileIntervalPeriod Attribute

The *ProfileIntervalPeriod* attribute is currently included in the *Get Profile Response* command payload, but does not appear in an attribute set. This represents the duration of each interval. *ProfileIntervalPeriod* represents the interval or time frame used to capture metered Energy, Gas, and Water consumption for profiling purposes. The enumeration for this field shall match one of the *ProfileIntervalPeriod* values defined in sub-clause D.3.2.3.1.1.1.[9]

D.3.2.2.1.17 PresetReadingTime

The *PresetReadingTime* attribute represents the time of day (in quarter hour increments) at which the meter will wake up and report a register reading even if there has been no consumption for the previous 24 hours. *PresetReadingTime* is an unsigned 16-bit value representing the hour and minutes. The byte usages are:

[9] CCB 1886

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Bits 0 to 7: Range of 0 to 0x3B representing the number of minutes past the top of the hour.

Bits 8 to 15: Range of 0 to 0x17 representing the hour of the day (in 24-hour format).

E.g.: A setting of 0x172D would represent 23:45 hours or 11:45 pm; a setting of 0x071E would represent 07:30 hours or 7:30 am. A setting of 0xFFFF indicates this feature is disabled. The use of Attribute Reporting Configuration is optional.

D.3.2.2.1.18 VolumePerReport Attribute

The *VolumePerReport* attribute represents the volume per report increment from the water or gas meter. For example a gas meter might be set to report its register reading for every time 1 cubic meter of gas is used. For a water meter it might report the register value every 10 liters of water usage.

D.3.2.2.1.19 FlowRestriction Attribute

The *FlowRestriction* attribute represents the volume per minute limit set in the flow restrictor. This applies to water but not for gas. A setting of 0xFF indicates this feature is disabled.

D.3.2.2.1.20 SupplyStatus Attribute

The *SupplyStatus* attribute represents the state of the supply at the customer's premises. The enumerated values for this field are outlined in Table D-13:

Table D-13– Supply Status Attribute Enumerations

Enumerated Value	Status
0x00	Supply OFF
0x01	Supply OFF/ARMED
0x02	Supply ON
0x03 to 0xFF	Reserved for future use

D.3.2.2.1.21 CurrentInletEnergyCarrierSummation Attribute

CurrentInletEnergyCarrierSummation is the current integrated volume of a given energy carrier measured on the inlet. The formatting and unit of measure for this value is specified in the *EnergyCarrierUnitOfMeasure* and *EnergyCarrierSummationFormatting* attributes (refer to Table D-25).

The Energy consumption registered in *CurrentSummationDelivered* is not necessarily a direct function of this value. The quality of the energy carrier may vary from day to day, e.g. Gas may have different quality.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

For heat and cooling meters the energy carrier is water at high or low temperature, the energy withdrawn from such a system is a function of the flow and the inlet and outlet temperature.

D.3.2.2.1.22 CurrentOutletEnergyCarrierSummation Attribute

CurrentOutletEnergyCarrierSummation is the current integrated volume of a given energy carrier measured on the outlet. The formatting and unit of measure for this value is specified in the *EnergyCarrierUnitOfMeasure* and *EnergyCarrierSummationFormatting* attributes (refer to Table D-25).

D.3.2.2.1.23 InletTemperature Attribute

InletTemperature is the temperature measured on the energy carrier inlet.

The formatting and unit of measure for this value is specified in the *TemperatureUnitOfMeasure* and *TemperatureFormatting* attributes (refer to Table D-25).

D.3.2.2.1.24 OutletTemperature Attribute

OutletTemperature is the temperature measured on the energy carrier outlet.

The formatting and unit of measure for this value is specified in the *TemperatureUnitOfMeasure* and *TemperatureFormatting* attributes (refer to Table D-25).

D.3.2.2.1.25 ControlTemperature Attribute

ControlTemperature is a reference temperature measured on the meter used to validate the Inlet/Outlet temperatures.

The formatting and unit of measure for this value is specified in the *TemperatureUnitOfMeasure* and *TemperatureFormatting* attributes (refer to Table D-25).

D.3.2.2.1.26 CurrentInletEnergyCarrierDemand Attribute

CurrentInletEnergyCarrierDemand is the current absolute demand on the energy carrier inlet.

The formatting and unit of measure for this value is specified in the *EnergyCarrierUnitOfMeasure* and *EnergyCarrierDemandFormatting* attributes (refer to Table D-25).

For a heat or cooling meter this will be the current absolute flow rate measured on the inlet.

D.3.2.2.1.27 CurrentOutletEnergyCarrierDemand Attribute

CurrentOutletEnergyCarrierDemand is the current absolute demand on the energy carrier outlet.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

The formatting and unit of measure for this value is specified in the *EnergyCarrierUnitOfMeasure* and *EnergyCarrierDemandFormatting* attributes (refer to Table D-25).

For a heat or cooling meter this will be the current absolute flow rate measured on the outlet.

D.3.2.2.1.28 PreviousBlockPeriodConsumptionDelivered Attribute

The *PreviousBlockPeriodConsumptionDelivered* attribute represents the total value of Energy, Gas or Water delivered and consumed in the premises at the end of the previous Block Tariff Period. If supported, the *PreviousBlockPeriodConsumptionDelivered* attribute is updated at the end of each Block Tariff Period.

D.3.2.2.1.29 CurrentBlockPeriodConsumptionReceived Attribute

The *CurrentBlockPeriodConsumptionReceived* attribute represents the most recent summed value of Energy, Gas or Water received by the energy supplier from the premises during the Block Tariff Period. The *CurrentBlockPeriodConsumptionReceived* attribute is reset at the start of each Block Tariff Period.

D.3.2.2.1.30 CurrentBlockReceived Attribute

When Block Tariffs are enabled, *CurrentBlockReceived* is an 8-bit Enumeration which indicates the currently active block. If blocks are active then the current active block is based on the *CurrentBlockPeriodConsumptionReceived* and the block thresholds. Block 1 is active when the value of *CurrentBlockPeriodConsumptionReceived* is less than or equal to the Block1Threshold value, Block 2 is active when *CurrentBlockPeriodConsumptionReceived* is greater than Block1Threshold value and less than or equal to the Block2Threshold value, and so on. Block 16 is active when the value of *CurrentBlockPeriodConsumptionReceived* is greater than Block15Threshold value. Refer to Table D-12 for block enumerations.

D.3.2.2.1.31 DFTSummationReceived Attribute

DFTSummationReceived represents a snapshot of attribute *CurrentSummationReceived* captured at the time indicated by the *DailyFreezeTime* attribute (see D.3.2.2.1.6).

If optionally provided, *DFTSummationReceived* is updated once every 24 hours and captured at the time set in the *DailyFreezeTime* attribute (see D.3.2.2.1.6).

D.3.2.2.1.32 ActiveRegisterTierDelivered Attribute

The ActiveRegisterTierDelivered attribute indicates the current register tier that the energy consumed is being accumulated against. Valid values for this attribute are defined in Table D-98.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

5563 D.3.2.2.1.33 ActiveRegisterTierReceived Attribute

5564 The *ActiveRegisterTierReceived* attribute indicates the current register tier that the energy
5565 generated is being accumulated against. Valid values for this attribute are defined in Table
5566 D-101.

5567 D.3.2.2.1.34 LastBlockSwitchTime Attribute

5568 This attribute allows other devices to determine the time at which a meter switches from one
5569 block to another.

5570 When Block Tariffs are enabled, the *LastBlockSwitchTime* attribute represents the
5571 timestamp of the last update to the *CurrentBlock* attribute, as a result of the consumption
5572 exceeding a threshold, or the start of a new block period and/or billing period.

5573 If, at the start of a new block period and/or billing period, the value of the *CurrentBlock* attribute
5574 is still set to Block1 (0x01), the *CurrentBlock* attribute value will not change but the
5575 *LastBlockSwitchTime* attribute shall be updated to indicate this change.
5576

5577 D.3.2.2.2 **Summation TOU Information Set**

5578 The following set of attributes provides a remote access to the Electric, Gas, or Water metering
5579 device's Time of Use (TOU) readings.

5580

5581 **Table D-14– TOU Information Attribute Set**

Identifier	Name	Type	Range	Access	Default	Man. /Opt.
0x0100	*CurrentTier1 SummationDelivered*	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0101	*CurrentTier1 SummationReceived*	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0102	*CurrentTier2 SummationDelivered*	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0103	*CurrentTier2 SummationReceived*	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0104	*CurrentTier3 SummationDelivered*	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFF	Read Only	-	O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x0105	*CurrentTier3 SummationReceived*	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0106	*CurrentTier4 SummationDelivered*	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0107	*CurrentTier4 SummationReceived*	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0108	*CurrentTier5 SummationDelivered*	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0109	*CurrentTier5 SummationReceived*	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x010A	*CurrentTier6 SummationDelivered*	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x010B	*CurrentTier6 SummationReceived*	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x010C	*CurrentTier7 SummationDelivered*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x010D	*CurrentTier7 SummationReceived*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x010E	*CurrentTier8 SummationDelivered*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x010F	*CurrentTier8 SummationReceived*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0110	*CurrentTier9 SummationDelivered*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0111	*CurrentTier9 SummationReceived*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0112	*CurrentTier10 SummationDelivered*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0113	*CurrentTier10 SummationReceived*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0114	*CurrentTier11 SummationDelivered*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x0115	*CurrentTier11 SummationReceived*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0116	*CurrentTier12 SummationDelivered*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0117	*CurrentTier12 SummationReceived*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0118	*CurrentTier13 SummationDelivered*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0119	*CurrentTier13 SummationReceived*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x011A	*CurrentTier14 SummationDelivered*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x011B	*CurrentTier14 SummationReceived*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x011C	*CurrentTier15 SummationDelivered*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x011D	*CurrentTier15 SummationReceived*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x011E	*CurrentTier16 SummationDelivered*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x011F	*CurrentTier16 SummationReceived*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0120	*CurrentTier17 SummationDelivered*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0121	*CurrentTier17 SummationReceived*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
…	…	…	…	…	…	…
0x015E	*CurrentTier48 SummationDelivered*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x015F	*CurrentTier48 SummationReceived*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x0160 to 0x01FB	Reserved					
0x01FC	*CPP1 Summation Delivered*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x01FD	Reserved					
0x01FE	*CPP2 Summation Delivered*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x01FF	Reserved					

5582

5583 D.3.2.2.2.1 CurrentTierNSummationDelivered Attributes

5584 Attributes *CurrentTier1SummationDelivered* through *CurrentTierNSummationDelivered*
5585 represent the most recent summed value of Energy, Gas, or Water delivered to the premises
5586 (i.e. delivered to the customer from the utility) at a specific price tier as defined by a TOU
5587 schedule or a real time pricing period. If optionally provided, attributes
5588 *CurrentTier1SummationDelivered* through *CurrentTierNSummationDelivered* are updated
5589 continuously as new measurements are made.

5590 D.3.2.2.2.2 CurrentTierNSummationReceived Attributes

5591 Attributes *CurrentTier1SummationReceived* through *CurrentTierNSummationReceived* represent
5592 the most recent summed value of Energy, Gas, or Water provided by the premises (i.e. received
5593 by the utility from the customer) at a specific price tier as defined by a TOU schedule or a real
5594 time pricing period. If optionally provided, attributes *CurrentTier1SummationReceived* through
5595 *CurrentTierNSummationReceived* are updated continuously as new measurements are made.

5596 D.3.2.2.2.3 CPP1SummationDelivered Attribute

5597 *CPP1SummationDelivered* represents the most recent summed value of Energy, Gas, or Water
5598 delivered to the premises (i.e. delivered to the customer from the utility) while Critical Peak
5599 Price 'CPP1' was being applied. If optionally provided, attribute *CPP1SummationDelivered* is
5600 updated continuously as new measurements are made.

5601 D.3.2.2.2.4 CPP2SummationDelivered Attribute

5602 *CPP2SummationDelivered* represents the most recent summed value of Energy, Gas, or Water
5603 delivered to the premises (i.e. delivered to the customer from the utility) while Critical Peak
5604 Price 'CPP2' was being applied. If optionally provided, attribute *CPP2SummationDelivered* is
5605 updated continuously as new measurements are made.

5606

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

5607 D.3.2.2.3 **Meter Status Attribute Set**

5608 The Meter Status Attribute Set is defined in Table D-15.

5609
5610 **Table D-15– Meter Status Attribute Set**

Identifier	Name	Type	Range	Access	Default	Man. / Opt.
0x0200	*Status*	8-bit BitMap	0x00 to 0xFF	Read Only	0x00	M
0x0201	*Remaining BatteryLife*	Unsigned 8-bit Integer	0x00 to 0xFF	Read Only	-	O
0x0202	*HoursIn Operation*	Unsigned 24bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O[10]
0x0203	*HoursIn Fault*	Unsigned 24bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0204	*Extended Status*	64-bit BitMap	0x0000000000000000 to 0xFFFFFFFFFFFFFFFF	Read Only	-	O
0x0205	*Remaining BatteryLife in Days*	Unsigned 16bit Integer	0x0000 to 0xFFFF	Read Only	-	O
0x0206	*CurrentMeter ID*	Octet String		Read Only	-	O
0x0207	*Ambient Consumption Indicator*	8-bit Enumeration	0x00 – 0x02	Read Only	-	O
0x0208-0x02FF	Reserved					

5611

5612 D.3.2.2.3.1 Status Attribute

5613 The *Status* attribute provides indicators reflecting the current error conditions found by the
5614 metering device. This attribute is an 8-bit field where when an individual bit is set, an error or
5615 warning condition exists. The behavior causing the setting or resetting each bit is device
5616 specific. In other words, the application within the metering device will determine and control
5617 when these settings are either set or cleared. Depending on the commodity type, the bits of this
5618 attribute will take on different meaning. Tables D.16, D.17, D.18, and D.19 below show the bit
5619 mappings for the *Status* attribute for Electricity, Gas, Water and Heating/Cooling
5620 respectively. A battery-operated meter will report any change in state of the *Status* when it
5621 wakes up via a ZCL report attributes command. The ESI is expected to make alarms

[10] CCB 1999

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

available to upstream systems together with consumption data collected from the battery operated meter.

Table D-16– Mapping of the *Status* Attribute (Electricity)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved	Service Disconnect Open	Leak Detect	Power Quality	Power Failure	Tamper Detect	Low Battery	Check Meter

The definitions of the Electricity *Status* bits are:

Service Disconnect Open: Set to true when the service has been disconnected to this premises.

Leak Detect: Set to true when a leak has been detected.

Power Quality: Set to true if a power quality event has been detected such as a low voltage, high voltage.

Power Failure: Set to true during a power outage.

Tamper Detect: Set to true if a tamper event has been detected.

Low Battery: Set to true when the battery needs maintenance.

Check Meter: Set to true when a non fatal problem has been detected on the meter such as a measurement error, memory error, self check error.

Table D-17– Meter *Status* Attribute (Gas)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reverse Flow	Service Disconnect	Leak Detect	Low Pressure	Not Defined	Tamper Detect	Low Battery	Check Meter

The definitions of the Gas *Status* bits are:

Reverse Flow: Set to true if flow detected in the opposite direction to normal (from consumer to supplier).

Service Disconnect: Set to true when the service has been disconnected to this premises. Ex. The valve is in the closed position preventing delivery of gas.

Leak Detect: Set to true when a leak has been detected.

Low Pressure: Set to true when the pressure at the meter is below the meter's low pressure

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

threshold value.

Tamper Detect: Set to true if a tamper event has been detected.

Low Battery: Set to true when the battery needs maintenance.

Check Meter: Set to true when a non fatal problem has been detected on the meter such as a measurement error, memory error, or self check error.

Table D-18– Meter *Status* Attribute (Water)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reverse Flow	Service Disconnect	Leak Detect	Low Pressure	Pipe Empty	Tamper Detect	Low Battery	Check Meter

The definitions of the Water *Status* bits are:

Reverse Flow: Set to true if flow detected in the opposite direction to normal (from consumer to supplier).

Service Disconnect: Set to true when the service has been disconnected to this premises. Ex. The valve is in the closed position preventing delivery of water.

Leak Detect: Set to true when a leak has been detected.

Low Pressure: Set to true when the pressure at the meter is below the meter's low pressure threshold value.

Pipe Empty: Set to true when the service pipe at the meter is empty and there is no flow in either direction.

Tamper Detect: Set to true if a tamper event has been detected.

Low Battery: Set to true when the battery needs maintenance.

Check Meter: Set to true when a non fatal problem has been detected on the meter such as a measurement error, memory error, or self check error.

Table D-19– Meter *Status* Attribute (Heat and Cooling)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Flow Sensor	Service Disconnect	Leak Detect	Burst Detect	Temperature Sensor	Tamper Detect	Low Battery	Check Meter[b]

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

The definitions of the Heat and Cooling *Status* bits are:

Flow Sensor: Set to true when an error is detected on a flow sensor at this premises.

Service Disconnect: Set to true when the service has been disconnected to this premises. Ex. The valve is in the closed position preventing delivery of heat or cooling.

Leak Detect: Set to true when a leak has been detected.

Burst Detect: Set to true when a burst is detected on pipes at this premises.

Temperature Sensor: Set to true when an error is detected on a temperature sensor at this premises.

Tamper Detect: Set to true if a tamper event has been detected.

Low Battery: Set to true when the battery needs maintenance.

Check Meter: Set to true when a non fatal problem has been detected on the meter such as a measurement error, memory error, or self check error.

Note: *It is not necessary to set aside Bit 7 as an "Extension Bit" for future expansion. If extra status bits are required an Extended Meter Status attribute may be added to support additional status values.*

D.3.2.2.3.2 RemainingBatteryLife Attribute

RemainingBatteryLife represents the estimated remaining life of the battery in % of capacity. A setting of 0xFF indicates this feature is disabled. The range 0 - 100 where 100 = 100%, 0xFF = Unknown.

D.3.2.2.3.3 HoursInOperation Attribute

HoursInOperation is a counter that increments once every hour during operation. This may be used as a check for tampering.

Note: *For meters that are not electricity meters turning off the meter does not necessarily prevent delivery of energy — but the meter might not be able to measure it.*

D.3.2.2.3.4 HoursInFault Attribute

HoursInFault is a counter that increments once every hour when the device is in operation with a fault detected. This may be used as a check for tampering.

Note: *For meters that are not electricity meters turning off the meter does not necessarily prevent delivery of energy - but the meter might not be able to measure it.*

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

5697 D.3.2.2.3.5 ExtendedStatus Attribute

5698 The ExtendedStatus attribute reflects the state of items in a meter that the standard Status
5699 attribute cannot show. The Extended Status BitMap is split into two groups of flags: general flags
5700 and metering type specific flags. Flags are currently defined for electricity and gas meters; flag
5701 definitions for other commodities will be added as and when their usage is agreed.

5702 These flags are set and reset by the meter autonomously; they cannot be reset by other devices.
5703 The mapping is as defined in the tables below. A meter which implements the attribute but does
5704 not implement a specific flag internally will simply have the corresponding bit always set to 0.
5705
5706 **Table D-20– General Flags of the Extended Status BitMap**

Bit	Flag name / Description
0	Meter Cover Removed
1	Strong Magnetic Field detected
2	Battery Failure
3	Program Memory Error
4	RAM Error
5	NV Memory Error
6	Measurement System Error
7	Watchdog Error
8	Supply Disconnect Failure
9	Supply Connect Failure
10	Measurement SW Changed/Tampered
11	Clock Invalid
12	Temperature Exceeded
13	Moisture Detected
14-23	Reserved

5707 The definitions of the General *Extended Status* bits are:

5708 **Meter Cover Removed:** Set to true when the device detects the meter cover being removed.

5709 **Strong Magnetic Field detected:** Set to true when the device detects presence of a strong
5710 magnetic field.

5711 **Battery Failure:** Set to true when the device detects that its battery has failed.

5712 **Program Memory Error:** Set to true when the device detects an error within its program (non-
5713 volatile) memory.

5714 **RAM Error:** Set to true when the device detects an instance of a Random Access Memory
5715 (RAM) error within the device memory.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

NV Memory Error: Set to true when the device detects an instance of a Non Volatile (NV) memory error within the device memory - this is a fatal meter error that will require the meter replacement.

Measurement System Error: Set to true when the device detects an error within its measurement system.

Watchdog Error: Set to true when the device has detected an instance of a watchdog reset event (following a catastrophic fault within the device).

Supply Disconnect Failure: Set to true when the device has detected that the valve has not closed as expected (for gas) or the contactor has not opened as expected (for electricity).

Supply Connect Failure: Set to true when the device has detected that the valve has not opened as expected (for gas) or the contactor has not closed as expected (for electricity).

Measurement SW Changed/Tampered: Set to true when the device detects that its measurement software has changed.

Clock Invalid: Set to true when the device detects that its internal clock is invalid.

Temperature Exceeded: Set to true when the metering device's temperature exceeds a predefined limit. There are various reasons for temperature rise in metering devices.

Moisture Detected: Set to true when a sensor has detected the presence of moisture e.g. moisture in a gas line which can cause a drop in gas pressure, or moisture detected in the sealed component area within a water meter.

Table D-21– Electricity -Meter specific Flags of the Extended Status BitMap

Bit	Flag name / Description
24	Terminal Cover Removed
25	Incorrect Polarity
26	Current with No Voltage
27	Limit Threshold Exceeded
28	Under Voltage
29	Over Voltage
30-63	Reserved

The definitions of the Electricity-Meter-Specific *Extended Status* bits are:

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Terminal Cover Removed: Set to true when the device detects that its terminal cover has been removed.

Incorrect Polarity: Set to true when the electricity meter detects incorrect polarity on the electricity supply.

Current with No Voltage: Set to true when the meter has been tampered with, to disconnect the measurement function from the supply. Electricity is still flowing but not being recorded.

Limit Threshold Exceeded: Set to true when the electricity meter detects that the load has exceeded the load limit threshold.

Under Voltage: Set to true when the electricity meter indicates that the voltage measurement over the voltage measurement period is lower than the voltage threshold.

Over Voltage: Set to true when the electricity meter indicates that the voltage measurement over the voltage measurement period is higher than the voltage threshold.

Table D-22– Gas-Meter specific Flags of the Extended Status BitMap

Bit	Flag name / Description
24	Battery Cover Removed
25	Tilt Tamper
26	Excess Flow
27-63	Reserved

The definitions of the Gas-Meter-Specific *Extended Status* bits are:

Battery Cover Removed: Set to true when the gas meter detects that its battery cover has been removed.

Tilt Tamper: Set to true when the meter detects a change in its physical properties (i.e. that it is being tilted, the tilt sensor has been activated or otherwise tampered with).

Excess Flow: Set to true when the gas meter detects excess flow (e.g. when local supply restoration is attempted).

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

5762 D.3.2.2.3.6 *RemainingBatteryLifeinDays* Attribute

5763 *RemainingBatteryLifeInDays* attribute represents the estimated remaining life of the battery in
5764 days of capacity. The range is 0 – 0xFFFE, where 0xFFFF represents 'Invalid', 'Unused' and
5765 'Disabled'.

5766 D.3.2.2.3.7 *CurrentMeterID* Attribute

5767 *CurrentMeterID* attribute is the current id for the Meter. This could be the current firmware
5768 version supported on the meter.

5769 D.3.2.2.3.8 *AmbientConsumptionIndicator* Attribute

5770 The *AmbientConsumptionIndicator* attribute is an 8-bit enumeration which provides a simple (i.e.
5771 Low/Medium/High) indication of the amount of a commodity being consumed within the
5772 premises. The status is achieved by comparing the current value of the *InstantaneousDemand*
5773 attribute (see D.3.2.2.5.1) with low/medium and medium/high thresholds. The status is defined in
5774 Table D-23:

5775 **Table D-23– LowMediumHighStatus Attribute**

Enumeration	Description
0x00	Low Energy usage
0x01	Medium Energy usage
0x02	High Energy usage

5776 The thresholds which are used to determine the value of this attribute are themselves defined as
5777 attributes within section D.10.2.2.4.1 and section D.10.2.2.4.2

5778

5779 D.3.2.2.4 **Formatting**

5780 The following set of attributes provides the ratios and formatting hints required to transform the
5781 received summations, consumptions, temperatures, or demands/ rates into displayable values.
5782 If the Multiplier and Divisor attribute values are non-zero, they are used in conjunction with the
5783 *SummationFormatting*, *ConsumptionFormatting*, *DemandFormatting*, and
5784 *TemperatureFormatting* attributes.

5785 Equations required to accomplish this task are defined below:

5786 Summation = Summation received * Multiplier / Divisor
5787 (formatted using *SummationFormatting*)

5788 Consumption = Summation received * Multiplier / Divisor
5789 (formatted using *ConsumptionFormatting*)

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Demand = Demand received * Multiplier / Divisor
(formatted using *DemandFormatting*)

Temperature = Temperature received * Multiplier / Divisor

If the Multiplier and Divisor attribute values are zero, just the formatting hints defined in *SummationFormatting*, *ConsumptionFormatting*, *DemandFormatting* and *TemperatureFormatting* attributes are used.

The summation received, consumption received, demand received, and temperature received variables used above can be replaced by any of the attributes listed in sub-clauses D.3.2.2.4.4, D.3.2.2.4.5, D.3.2.2.4.6, D.3.2.2.4.11, D.3.2.2.4.12, and D.3.2.2.4.14.

The following table shows examples that demonstrate the relation between these attributes.

Table D-24– Formatting Examples

Attribute	Example 1	Example 2	Example 3
Value as transmitted and received	52003	617	23629
UnitofMeasure	kWh	CCF	kWh
Multiplier	1	2	6
Divisor	1000	100	10000
Number of Digits to the left of the Decimal Point	5	4	5
Number of Digits to the right of the Decimal Point	0	2	3
Suppress leading zeros	False	False	True
Displayed value	00052	0012.34	14.177

The Consumption Formatting Attribute Set is defined in Table D-25.

Table D-25– Formatting Attribute Set

Identifier	Name	Type	Range	Access	Default	Man./ Opt.
0x0300	*UnitofMeasure*	8-bit Enumeration	0x00 to 0xFF	Read Only	0x00	M
0x0301	*Multiplier*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x0302	*Divisor*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0303	*SummationFormatting*	8-bit BitMap	0x00 to 0xFF	Read Only	-	M
0x0304	*DemandFormatting*	8-bit BitMap	0x00 to 0xFF	Read Only	-	O
0x0305	*HistoricalConsumptionFormatting*	8-bit BitMap	0x00 to 0xFF	Read Only	-	O
0x0306	*MeteringDeviceType*	8-bit BitMap	0x00 to 0xFF	Read Only	-	M
0x0307	*SiteID*	Octet String	1 to 33 Octets	Read only	-	O
0x0308	*MeterSerialNumber*	Octet String	1 to 25 Octets	Read only	-	O
0x0309	*EnergyCarrierUnit OfMeasure*	8-bit Enumeration	0x00 to 0xFF	Read Only	-	O[11]
0x030A	*EnergyCarrier SummationFormatting*	8-bit BitMap	0x00 to 0xFF	Read Only	-	O[12]
0x030B	*EnergyCarrier DemandFormatting*	8-bit BitMap	0x00 to 0xFF	Read Only	-	O
0x030C	*TemperatureUnit OfMeasure*	8-bit Enumeration	0x00 to 0xFF	Read Only	-	O[13]
0x030D	*TemperatureFormatting*	8-bit BitMap	0x00 to 0xFF	Read Only	-	O[14]
0x030E	*ModuleSerialNumber*	Octet String	1 to 25 Octets	Read only	-	O
0x030F	*OperatingTariffLabel Delivered*	Octet String	1 to 25 Octets	Read only	-	O
0x0310	*OperatingTariffLabel Received*	Octet String	1 to 25 Octets	Read only	-	O
0x0311	*CustomerIDNumber*	Octet String	1 to 25 Octets	Read only	-	O
0x0312	*AlternativeUnitof Measure*	8-bit Enumeration	0x00 to 0xFF	Read Only	0x00	O

[11] CCB 1999
[12] CCB 1999
[13] CCB 1999
[14] CCB 1999

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x0313	*Alternative DemandFormatting*	8-bit BitMap	0x00 to 0xFF	Read Only	-	O
0x0314	*Alternative ConsumptionFormatting*	8-bit BitMap	0x00 to 0xFF	Read Only	-	O
0x0313 to 0x03FF	Reserved					

5805

5806 D.3.2.2.4.1 UnitofMeasure Attribute

5807 *UnitofMeasure* provides a label for the Energy, Gas, or Water being measured by the metering
5808 device. The unit of measure applies to all summations, consumptions/ profile interval and
5809 demand/rate supported by this cluster other than those specifically identified as being based upon
5810 the *EnergyCarrierUnitOfMeasure* or the *AlternativeUnitofMeasure*. Other measurements such as
5811 the power factor are self describing. This attribute is an 8-bit enumerated field. The bit
5812 descriptions for this Attribute are listed in Table D-26.

5813

5814 **Table D-26– *UnitofMeasure* Attribute Enumerations**

Values	Description
0x00	kWh (Kilowatt Hours) & kW (Kilowatts) in pure binary format
0x01	m^3 (Cubic Meter) & m^3/h (Cubic Meter per Hour) in pure binary format
0x02	ft^3 (Cubic Feet) & ft^3/h (Cubic Feet per Hour) in pure binary format
0x03	ccf ((100 or Centum) Cubic Feet) & ccf/h ((100 or Centum) Cubic Feet per Hour) in pure binary format
0x04	US gl (US Gallons) & US gl/h (US Gallons per Hour) in pure binary format.
0x05	IMP gl (Imperial Gallons) & IMP gl/h (Imperial Gallons per Hour) in pure binary format
0x06	BTUs & BTU/h in pure binary format
0x07	Liters & l/h (Liters per Hour) in pure binary format
0x08	kPA (gauge) in pure binary format
0x09	kPA (absolute) in pure binary format
0x0A	mcf (1000 Cubic Feet) & mcf/h (1000 Cubic feet per hour) in pure binary format
0x0B	Unitless in pure binary format
0x0C	MJ (Mega Joule) and MJ/s (Mega Joule per second (MW)) in pure binary format
0x0D	kVar & kVarh in Binary Format
0x0E to 0x7F	Reserved for future use.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x80	kWh (Kilowatt Hours) & kW (Kilowatts) in BCD format
0x81	m^3 (Cubic Meter) & m^3/h (Cubic Meter per Hour) in BCD format
0x82	ft^3 (Cubic Feet) & ft^3/h (Cubic Feet per Hour) in BCD format
0x83	ccf ((100 or Centum) Cubic Feet) & ccf/h ((100 or Centum) Cubic Feet per Hour) in BCD format
0x84	US gl (US Gallons) & US gl/h (US Gallons per Hour) in BCD format
0x85	IMP gl (Imperial Gallons) & IMP gl/h (Imperial Gallons per Hour) in BCD Format
0x86	BTUs & BTU/h in BCD format
0x87	Liters & l/h (Liters per Hour) in BCD format
0x88	kPA (gauge) in BCD format
0x89	kPA (absolute) in BCD format
0x8A	mcf (1000 Cubic Feet) & mcf/h (1000 Cubic Feet per Hour) in BCD format
0x8B	unitless in BCD format
0x8C	MJ (Mega Joule) and MJ/s (Mega Joule per second (MW)) in BCD format
0x8D	kVar & kVarh in BCD Format
0x8E to 0xFF	Reserved for future use.

Note: *When using BCD for meter reads, the values A to F are special values or indicators denoting "Opens", "Shorts", and etc. conditions when reading meter register hardware. Any SE device displaying the BCD based values to end users should use a non-decimal value to replace the A to F. In other words, a device could use an "*" in place of the special values or indicators.*

D.3.2.2.4.2 Multiplier Attribute

Multiplier provides a value to be multiplied against a raw or uncompensated sensor count of Energy, Gas, or Water being measured by the metering device. If present, this attribute must be applied against all summation, consumption and demand values to derive the delivered and received values expressed in the unit of measure specified. This attribute must be used in conjunction with the *Divisor* attribute.

D.3.2.2.4.3 Divisor Attribute

Divisor provides a value to divide the results of applying the Multiplier Attribute against a raw or uncompensated sensor count of Energy, Gas, or Water being measured by the metering device. If present, this attribute must be applied against all summation, consumption and demand values to derive the delivered and received values expressed in the unit of measure specified. This attribute must be used in conjunction with the *Multiplier* attribute.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

5833 D.3.2.2.4.4 SummationFormatting Attribute

5834 *SummationFormatting* provides a method to properly decipher the number of digits and the
5835 decimal location of the values found in the Summation Information Set of attributes. This
5836 attribute is to be decoded as follows:

5837 **Bits 0 to 2:** Number of Digits to the right of the Decimal Point.

5838 **Bits 3 to 6:** Number of Digits to the left of the Decimal Point.

5839 **Bit 7:** If set, suppress leading zeros.

5840 This attribute shall be used against the following attributes:

5841 • *CurrentSummationDelivered*

5842 • *CurrentSummationReceived*

5843 • TOU Information attributes

5844 • *DFTSummation*

5845 • Block Information attributes

5846 D.3.2.2.4.5 DemandFormatting Attribute

5847 *DemandFormatting* provides a method to properly decipher the number of digits and the
5848 decimal location of the values found in the Demand-related attributes. This attribute is to be
5849 decoded as follows:

5850 **Bits 0 to 2:** Number of Digits to the right of the Decimal Point.

5851 **Bits 3 to 6:** Number of Digits to the left of the Decimal Point.

5852 **Bit 7:** If set, suppress leading zeros.

5853 This attribute shall be used against the following attributes:

5854 • *CurrentMaxDemandDelivered*

5855 • *CurrentMaxDemandReceived*

5856 • *InstantaneousDemand*

5857 D.3.2.2.4.6 HistoricalConsumptionFormatting Attribute

5858 *HistoricalConsumptionFormatting* provides a method to properly decipher the number of digits
5859 and the decimal location of the values found in the Historical Consumption Set of attributes. This
5860 attribute is to be decoded as follows:

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Bits 0 to 2: Number of Digits to the right of the Decimal Point.

Bits 3 to 6: Number of Digits to the left of the Decimal Point.

Bit 7: If set, suppress leading zeros.

This attribute shall be used against the following attributes:

- *CurrentDayConsumptionDelivered*
- *CurrentDayConsumptionReceived*
- *PreviousDayConsumptionDelivered*
- *PreviousDayConsumptionReceived*
- *CurrentPartialProfileIntervalValue*
- *Intervals*
- *DailyConsumptionTarget*
- *CurrentDayConsumptionDelivered*
- *CurrentDayConsumptionReceived*
- *PreviousDayNConsumptionDelivered*
- *PreviousDayNConsumptionReceived*
- *CurrentWeekConsumptionDelivered*
- *CurrentWeekConsumptionReceived*
- *PreviousWeekNConsumptionDelivered*
- *PreviousWeekNConsumptionReceived*
- *CurrentMonthConsumptionDelivered*
- *CurrentMonthConsumptionReceived*
- *PreviousMonthNConsumptionDelivered*
- *PreviousMonthNConsumptionReceived*

D.3.2.2.4.7 MeteringDeviceType Attribute

MeteringDeviceType provides a label for identifying the type of metering device present. The attribute are values representing Energy, Gas, Water, Thermal, Heat, Cooling, and mirrored metering devices. The defined values are represented in Table D-27. (Note that these values represent an Enumeration, and not an 8-bit BitMap as indicated in the attribute description.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

5889 For backwards compatibility reasons, the data type has not been changed, though the data
5890 itself should be treated like an enum.)

5891 Where a mirror is provided for a battery-powered metering device, the mirror shall assume the
5892 relevant 'Mirrored Metering' device type (127-139) whilst the meter itself shall utilize the
5893 'Metering' device type (0 to 15). It shall be the responsibility of the device providing the
5894 mirror to modify the Device Type shown on the mirror to that of a 'Mirrored Metering' device.

5895 **Table D-27– *MeteringDeviceType* Attribute**

Values	Description
0	Electric Metering
1	Gas Metering
2	Water Metering
3	Thermal Metering (deprecated)
4	Pressure Metering
5	Heat Metering
6	Cooling Metering
7	End Use Measurement Device (EUMD) for metering electric vehicle charging
8	PV Generation Metering
9	Wind Turbine Generation Metering
10	Water Turbine Generation Metering
11	Micro Generation Metering
12	Solar Hot Water Generation Metering
13	Electric Metering Element/Phase 1
14	Electric Metering Element/Phase 2
15	Electric Metering Element/Phase 3
16 to 126	Reserved for future growth
127	Mirrored Electric Metering
128	Mirrored Gas Metering
129	Mirrored Water Metering
130	Mirrored Thermal Metering (deprecated)
131	Mirrored Pressure Metering
132	Mirrored Heat Metering
133	Mirrored Cooling Metering
134	Mirrored End Use Measurement Device (EUMD) for metering electric vehicle charging

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

135	Mirrored PV Generation Metering
136	Mirrored Wind Turbine Generation Metering
137	Mirrored Water Turbine Generation Metering
138	Mirrored Micro Generation Metering
139	Mirrored Solar Hot Water Generation Metering
140	Mirrored Electric Metering Element/Phase 1
141	Mirrored Electric Metering Element/Phase 2
142	Mirrored Electric Metering Element/Phase 3
143 to 255	Reserved for future growth

Note: *Heat and cooling meters are used for measurement and billing of heat (and cooling) delivered through liquid (water) based central heating systems. The consumers are typically billed by the kWh, calculated from the flow and the temperatures in and out.*

D.3.2.2.4.8 SiteID Attribute

The *SiteID* is a ZCL Octet String field capable of storing a 32 character string (the first Octet indicates length) encoded in UTF-8 format. The *SiteID* is a text string, known in the UK as the MPAN number for electricity, MPRN for gas and 'Stand Point' in South Africa. These numbers specify the meter point location in a standardized way. The field is defined to accommodate the number of characters typically found in the UK and Europe (16 digits). Generally speaking the field is numeric but is defined for the possibility of an alpha-numeric format by specifying an octet string.

D.3.2.2.4.9 MeterSerialNumber Attribute

The *MeterSerialNumber* is a ZCL Octet String field capable of storing a 24 character string (the first Octet indicates length) encoded in UTF-8 format. It is used to provide a unique identification of the metering device.

D.3.2.2.4.10 EnergyCarrierUnitOfMeasure Attribute

The *EnergyCarrierUnitOfMeasure* specifies the unit of measure that the *EnergyCarrier* is measured in. This unit of measure is typically a unit of volume or flow and cannot be an amount of energy. The enumeration of this attribute is otherwise identical to the *UnitofMeasure* attribute (Table D-26).

D.3.2.2.4.11 EnergyCarrierSummationFormatting Attribute

EnergyCarrierSummationFormatting provides a method to properly decipher the number of digits and the decimal location of the values found in the Summation- related attributes.

This attribute is to be decoded as follows:

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Bits 0 to 2: Number of Digits to the right of the Decimal Point.

Bits 3 to 6: Number of Digits to the left of the Decimal Point.

Bit 7: If set, suppress leading zeros.

This attribute shall be used in relation with the following attributes:

- *CurrentInletEnergyCarrierSummation*
- *CurrentOutletEnergyCarrierSummation*

D.3.2.2.4.12 EnergyCarrierDemandFormatting Attribute

EnergyCarrierDemandFormatting provides a method to properly decipher the number of digits and the decimal location of the values found in the Demand- related attributes.

This attribute is to be decoded as follows:

Bits 0 to 2: Number of Digits to the right of the Decimal Point.

Bits 3 to 6: Number of Digits to the left of the Decimal Point.

Bit 7: If set, suppress leading zeros.

This attribute shall be used in relation with the following attributes:

- *CurrentInletEnergyCarrierDemand*
- *CurrentOutletEnergyCarrierDemand*
- *CurrentDayMaxEnergyCarrierDemand*
- *PreviousDayMaxEnergyCarrierDemand*
- *CurrentMonthMaxEnergyCarrierDemand*
- *CurrentMonthMinEnergyCarrierDemand*
- *CurrentYearMinEnergyCarrierDemand*
- *CurrentYearMaxEnergyCarrierDemand*

D.3.2.2.4.13 TemperatureUnitOfMeasure Attribute

The *TemperatureUnitOfMeasure* specifies the unit of measure that temperatures are measured in. The enumeration of this attribute is as follows.

Table D-28– *TemperatureUnitOfMeasure* Enumeration

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Values	Description
0x00	K (Degrees Kelvin) in pure Binary format.
0x01	°C (Degrees Celsius) in pure Binary format.
0x02	°F (Degrees Fahrenheit) in pure Binary format.
0x03-0x7F	Reserved for future use
0x80	K (Degrees Kelvin) in BCD format.
0x81	°C (Degrees Celsius) in BCD format.
0x82	°F (Degrees Fahrenheit) in BCD format.
0x83-0xFF	Reserved for future use

5946

5947 D.3.2.2.4.14 TemperatureFormatting Attribute

5948 *TemperatureFormatting* provides a method to properly decipher the number of digits and the
5949 decimal location of the values found in the Temperature-related attributes. This attribute is to
5950 be decoded as follows:

5951 **Bits 0 to 2:** Number of Digits to the right of the Decimal Point.

5952 **Bits 3 to 6:** Number of Digits to the left of the Decimal Point.

5953 **Bit 7:** If set, suppress leading zeros.

5954 This attribute shall be used in relation with the following attributes:

5955 • *InletTemperature*

5956 • *OutletTemperature*

5957 • *ControlTemperature*

5958 D.3.2.2.4.15 ModuleSerialNumber Attribute

5959 The *ModuleSerialNumber* attribute represents the serial number (unique identifier) of the meter
5960 module. It is a ZCL Octet String field capable of storing a 24 character string (the first Octet
5961 indicates length) encoded in UTF-8 format. It shall be used to uniquely identify the meter
5962 communications module.

5963 D.3.2.2.4.16 OperatingTariffLabelDelivered Attribute

5964 The *OperatingTariffLabelDelivered* attribute is the meter's version of the *TariffLabel* attribute
5965 that is found within the Tariff Information attribute set of the Price Cluster. It is used to identify
5966 the current consumption tariff operating on the meter. See section D.4.2.2.7.1. The attribute is a
5967 ZCL Octet String field capable of storing a 24 character string (the first Octet indicates
5968 length) encoded in UTF-8 format.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

D.3.2.2.4.17 OperatingTariffLabelReceived Attribute

The *OperatingTariffLabelReceived* attribute is the meter's version of the *ReceivedTariffLabel* attribute that is found within the Tariff Information attribute set of the Price Cluster. It is used to identify the current generation tariff operating on the meter. See section D.4.2.2.15.1. The attribute is a ZCL Octet String field capable of storing a 24 character string (the first Octet indicates length) encoded in UTF-8 format.

D.3.2.2.4.18 CustomerIDNumber Attribute

The *CustomerIDNumber* attribute provides a customer identification which may be used to confirm the customer at the premises. The attribute is a ZCL Octet String field capable of storing a 24 character string (not including the first Octet which indicates length) encoded in UTF-8 format.

D.3.2.2.4.19 AlternativeUnitofMeasure Attribute

Unless stated otherwise, the *AlternativeUnitofMeasure* attribute provides a base for the attributes in the Alternative Historical Consumption attribute set defined in Table D-46.

The *AlternativeUnitofMeasure* attribute shall be supported if any of the attributes within the Alternative Historical Consumption attribute set are to be used.

The *AlternativeUnitofMeasure* attribute shall be set to a value that is different to the *UnitOfMeasure* attribute.

The *AlternativeUnitofMeasure* attribute is an 8-bit enumerated field. The possible values for this attribute are listed in Table D-26.

D.3.2.2.4.20 AlternativeDemandFormatting Attribute

AlternativeDemandFormatting provides a method to properly decipher the number of digits and the decimal location of the values found in the Alternative Demand-related attributes. This attribute is to be decoded as follows:

Bits 0 to 2: Number of Digits to the right of the Decimal Point.

Bits 3 to 6: Number of Digits to the left of the Decimal Point.

Bit 7: If set, suppress leading zeros.

This attribute shall be used against the following attribute:

- *AlternativeInstantaneousDemand*

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

5999 D.3.2.2.4.21 AlternativeConsumptionFormatting Attribute

6000 *AlternativeConsumptionFormatting* provides a method to properly decipher the number of digits
6001 and the decimal location of the consumption values found in the Alternative Historical
6002 Consumption Set of attributes. This attribute is to be decoded as follows:

6003 **Bits 0 to 2:** Number of Digits to the right of the Decimal Point.

6004 **Bits 3 to 6:** Number of Digits to the left of the Decimal Point.

6005 **Bit 7:** If set, suppress leading zeros.

6006 This attribute shall be used against the following attributes:

6007 • *CurrentDayAlternativeConsumptionDelivered*
6008 • *CurrentDayAlternativeConsumptionReceived*
6009 • *PreviousDayAlternativeConsumptionDelivered*
6010 • *PreviousDayAlternativeConsumptionReceived*
6011 • *CurrentAlternativePartialProfileIntervalValue*
6012 • *PreviousDayNAlternativeConsumptionDelivered*
6013 • *PreviousDayNAlternativeConsumptionReceived*
6014 • *CurrentWeekAlternativeConsumptionDelivered*
6015 • *CurrentWeekAlternativeConsumptionReceived*
6016 • *PreviousWeekNAlternativeConsumptionDelivered*
6017 • *PreviousWeekNAlternativeConsumptionReceived*
6018 • *CurrentMonthAlternativeConsumptionDelivered*
6019 • *CurrentMonthAlternativeConsumptionReceived*
6020 • *PreviousMonthNAlternativeConsumptionDelivered*
6021 • *PreviousMonthNAlternativeConsumptionReceived*

6022

6023

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

6024 D.3.2.2.5 **Historical Consumption Attribute**

6025 The Historical Consumption attribute set allows historical information to be presented in a base
6026 defined by the *UnitofMeasure* attribute (see D.3.2.2.4.1). The attributes within this set are
6027 defined in Table D-29.

6028 **Table D-29– Historical Attribute Set**

Identifier	Name	Type	Range	Access	Default	Man./ Opt.
0x0400	*InstantaneousDemand*	Signed 24-bit Integer	-8,388,607 to 8,388,607	Read Only	0x00	O
0x0401	*CurrentDayConsumption Delivered*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0402	*CurrentDayConsumption Received*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0403	*PreviousDayConsumption Delivered*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0404	*PreviousDayConsumption Received*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0405	*CurrentPartialProfile IntervalStartTimeDelivered*	UTCTime		Read Only	-	O
0x0406	*CurrentPartialProfile IntervalStartTimeReceived*	UTCTime	0x000000 to 0xFFFFFF	Read Only	-	O
0x0407	*CurrentPartialProfile IntervalValueDelivered*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0408	*CurrentPartialProfile IntervalValueReceived*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0409	*CurrentDayMaxPressure*	Unsigned 48-bit Integer	0x000000 000000 to 0xFFFFFF FFFFFF	Read Only	-	O
0x040A	*CurrentDayMinPressure*	Unsigned 48-bit Integer	0x000000 000000 to 0xFFFFFF FFFFFF	Read Only	-	O
0x040B	*PreviousDayMaxPressure*	Unsigned 48-bit Integer	0x000000 000000 to 0xFFFFFF FFFFFF	Read Only	-	O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x040C	*PreviousDayMinPressure*	Unsigned 48-bit Integer	0x000000 000000 to 0xFFFFFF FFFFFF	Read Only	-	O
0x040D	*CurrentDayMaxDemand*	Signed 24-bit Integer	-8,388,607 to 8,388,607	Read Only	-	O
0x040E	*PreviousDayMaxDemand*	Signed 24-bit Integer	-8,388,607 to 8,388,607	Read Only	-	O
0x040F	*CurrentMonthMax Demand*	Signed 24-bit Integer	-8,388,607 to 8,388,607	Read Only	-	O
0x0410	*CurrentYearMaxDemand*	Signed 24-bit Integer	-8,388,607 to 8,388,607	Read Only	-	O
0x0411	*CurrentDayMaxEnergy CarrierDemand*	Signed 24-bit integer	-8,388,607 to 8,388,607	Read Only	-	O
0x0412	*PreviousDayMaxEnergy CarrierDemand*	Signed 24-bit integer	-8,388,607 to 8,388,607	Read Only	-	O
0x0413	*CurrentMonthMax EnergyCarrierDemand*	Signed 24-bit Integer	-8,388,607 to 8,388,607	Read Only	-	O
0x0414	*CurrentMonthMinEnergy CarrierDemand*	Signed 24-bit Integer	-8,388,607 to 8,388,607	Read Only	-	O
0x0415	*CurrentYearMaxEnergy CarrierDemand*	Signed 24-bit Integer	-8,388,607 to 8,388,607	Read Only	-	O
0x0416	*CurrentYearMinEnergy CarrierDemand*	Signed 24-bit Integer	-8,388,607 to 8,388,607	Read Only	-	O
0x0417 to 0x041F	Reserved					
0x0420	*PreviousDay2Consumption Delivered*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0421	*PreviousDay2Consumption Received*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0422	*PreviousDay3Consumption Delivered*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0423	*PreviousDay3Consumption Received*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x0424	*PreviousDay4ConsumptionDelivered*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0425	*PreviousDay4ConsumptionReceived*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0426	*PreviousDay5ConsumptionDelivered*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0427	*PreviousDay5ConsumptionReceived*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0428	*PreviousDay6ConsumptionDelivered*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0429	*PreviousDay6ConsumptionReceived*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x042A	*PreviousDay7ConsumptionDelivered*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x042B	*PreviousDay7ConsumptionReceived*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x042C	*PreviousDay8ConsumptionDelivered*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x042D	*PreviousDay8ConsumptionReceived*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x042E to 0x042F	Reserved					
0x0430	*CurrentWeekConsumptionDelivered*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0431	*CurrentWeekConsumptionReceived*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0432	*PreviousWeekConsumptionDelivered*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0433	*PreviousWeekConsumptionReceived*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x0434	*PreviousWeek2Consumption Delivered*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0435	*PreviousWeek2Consumption Received*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0436	*PreviousWeek3Consumption Delivered*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0437	*PreviousWeek3Consumption Received*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0438	*PreviousWeek4Consumption Delivered*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0439	*PreviousWeek4Consumption Received*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x043A	*PreviousWeek5Consumption Delivered*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x043B	*PreviousWeek5Consumption Received*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x043C to 0x043F	Reserved					
0x0440	*CurrentMonthConsumption Delivered*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0441	*CurrentMonthConsumption Received*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0442	*PreviousMonthConsumption Delivered*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0443	*PreviousMonthConsumption Received*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0444	*PreviousMonth2Consumption Delivered*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0445	*PreviousMonth2Consumption Received*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0446	*PreviousMonth3Consumption Delivered*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x0447	*PreviousMonth3ConsumptionReceived*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0448	*PreviousMonth4ConsumptionDelivered*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0449	*PreviousMonth4ConsumptionReceived*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x044A	*PreviousMonth5ConsumptionDelivered*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x044B	*PreviousMonth5ConsumptionReceived*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x044C	*PreviousMonth6ConsumptionDelivered*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x044D	*PreviousMonth6ConsumptionReceived*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x044E	*PreviousMonth7ConsumptionDelivered*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x044F	*PreviousMonth7ConsumptionReceived*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0450	*PreviousMonth8ConsumptionDelivered*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0451	*PreviousMonth8ConsumptionReceived*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0452	*PreviousMonth9ConsumptionDelivered*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0453	*PreviousMonth9ConsumptionReceived*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0454	*PreviousMonth10ConsumptionDelivered*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0455	*PreviousMonth10ConsumptionReceived*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0456	*PreviousMonth11ConsumptionDelivered*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x0457	*PreviousMonth11Consumption Received*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0458	*PreviousMonth12Consumption Delivered*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0459	*PreviousMonth12Consumption Received*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x045A	*PreviousMonth13Consumption Delivered*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x045B	*PreviousMonth13Consumption Received*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x045C	*Historical Freeze Time*	Unsigned 16-bit Integer	0x0000 to 0x173B	Read Only	0x0000	O
0x045D to 0x04FF	Reserved					

6029

6030 D.3.2.2.5.1 InstantaneousDemand Attribute

6031 *InstantaneousDemand* represents the current Demand of Energy, Gas, or Water delivered or
6032 received at the premises. Positive values indicate demand delivered to the premises where
6033 negative values indicate demand received from the premises. *InstantaneousDemand* is
6034 updated continuously as new measurements are made. The frequency of updates to this field is
6035 specific to the metering device, but should be within the range of once every second to once every
6036 5 seconds.

6037 D.3.2.2.5.2 CurrentDayConsumptionDelivered Attribute

6038 *CurrentDayConsumptionDelivered* represents the summed value of Energy, Gas, or Water
6039 delivered to the premises since the *Historical Freeze Time* (HFT). If optionally provided,
6040 *CurrentDayConsumptionDelivered* is updated continuously as new measurements are made. If
6041 the optional HFT attribute is not available, default to midnight local time.

6042 D.3.2.2.5.3 CurrentDayConsumptionReceived Attribute

6043 *CurrentDayConsumptionReceived* represents the summed value of Energy, Gas, or Water
6044 received from the premises since the *Historical Freeze Time* (HFT). If optionally provided,
6045 *CurrentDayConsumptionReceived* is updated continuously as new measurements are made. If
6046 the optional HFT attribute is not available, default to midnight local time.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

D.3.2.2.5.4 PreviousDayConsumptionDelivered Attribute

PreviousDayConsumptionDelivered represents the summed value of Energy, Gas, or Water delivered to the premises within the previous 24 hour period starting at the *Historical Freeze Time* (HFT). If optionally provided, *PreviousDayConsumptionDelivered* is updated every HFT. If the optional HFT attribute is not available, default to midnight local time.

D.3.2.2.5.5 PreviousDayConsumptionReceived Attribute

PreviousDayConsumptionReceived represents the summed value of Energy, Gas, or Water received from the premises within the previous 24 hour period starting at the *Historical Freeze Time* (HFT). If optionally provided, *PreviousDayConsumptionReceived* is updated every HFT. If the optional *HFT* attribute is not available, default to midnight local time.

D.3.2.2.5.6 CurrentPartialProfileIntervalStartTimeDelivered Attribute

CurrentPartialProfileIntervalStartTimeDelivered represents the start time of the current Load Profile interval being accumulated for commodity delivered.

D.3.2.2.5.7 CurrentPartialProfileIntervalStartTimeReceived Attribute

CurrentPartialProfileIntervalStartTimeReceived represents the start time of the current Load Profile interval being accumulated for commodity received.

D.3.2.2.5.8 CurrentPartialProfileIntervalValueDelivered Attribute

CurrentPartialProfileIntervalValueDelivered represents the value of the current Load Profile interval being accumulated for commodity delivered.

D.3.2.2.5.9 CurrentPartialProfileIntervalValueReceived Attribute

CurrentPartialProfileIntervalValueReceived represents the value of the current Load Profile interval being accumulated for commodity received.

D.3.2.2.5.10 CurrentDayMaxPressure Attribute

CurrentDayMaxPressure is the maximum pressure reported during a day from the water or gas meter.

D.3.2.2.5.11 PreviousDayMaxPressure Attribute

PreviousDayMaxPressure represents the maximum pressure reported during previous day from the water or gas meter.

D.3.2.2.5.12 CurrentDayMinPressure Attribute

CurrentDayMinPressure is the minimum pressure reported during a day from the water or gas meter.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

D.3.2.2.5.13 PreviousDayMinPressure Attribute

PreviousDayMinPressure represents the minimum pressure reported during previous day from the water or gas meter.

D.3.2.2.5.14 CurrentDayMaxDemand Attribute

CurrentDayMaxDemand represents the maximum demand or rate of delivered value of Energy, Gas, or Water being utilized at the premises.

D.3.2.2.5.15 PreviousDayMaxDemand Attribute

PreviousDayMaxDemand represents the maximum demand or rate of delivered value of Energy, Gas, or Water being utilized at the premises.

Note: *At the end of a day the metering device will transfer the CurrentDayMaxPressure into PreviousDayMaxPressure, CurrentDayMinPressure into PreviousDayMinPressure and CurrentDayMaxDemand into PreviousDayMaxDemand.*

D.3.2.2.5.16 CurrentMonthMaxDemand Attribute

CurrentMonthMaxDemand is the maximum demand reported during a month from the meter.

For electricity, heat and cooling meters this is the maximum power reported in a month.

D.3.2.2.5.17 CurrentYearMaxDemand Attribute

CurrentYearMaxDemand is the maximum demand reported during a year from the meter.

For electricity, heat and cooling meters this is the maximum power reported in a year.

D.3.2.2.5.18 CurrentDayMaxEnergyCarrierDemand Attribute

CurrentDayMaxEnergyCarrierDemand is the maximum energy carrier demand reported during a day from the meter.

Note: *At the end of a day the meter will transfer the CurrentDayMaxEnergyCarrierDemand into PreviousDayMaxEnergyCarrierDemand.*

For heat and cooling meters this is the maximum flow rate on the inlet reported in a day.

D.3.2.2.5.19 PreviousDayMaxEnergyCarrierDemand Attribute

PreviousDayMaxEnergyCarrierDemand is the maximum energy carrier demand reported during the previous day from the meter.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

D.3.2.2.5.20 CurrentMonthMaxEnergyCarrierDemand Attribute

CurrentMonthMaxEnergyCarrierDemand is the maximum energy carrier demand reported during a month from the meter.

For heat and cooling meters this is the maximum flow rate on the inlet reported in a month.

D.3.2.2.5.21 CurrentMonthMinEnergyCarrierDemand Attribute

CurrentMonthMinEnergyCarrierDemand is the minimum energy carrier demand reported during a month from the meter.

For heat and cooling meters this is the minimum flow rate on the inlet reported in a month.

Note: *This attribute may be used to detect leaks if there has been no flow rate of zero in the last month.*

D.3.2.2.5.22 CurrentYearMaxEnergyCarrierDemand Attribute

CurrentYearMaxEnergyCarrierDemand is the maximum energy carrier demand reported during a year from the meter.

For heat and cooling meters this is the maximum flow rate on the inlet reported in a year.

D.3.2.2.5.23 CurrentYearMinEnergyCarrierDemand Attribute

CurrentYearMinEnergyCarrierDemand is the minimum energy carrier demand reported during a year from the heat meter.

For heat and cooling meters this is the minimum flow rate on the inlet reported in a year.

Note: *This attribute may be used to detect leaks if there has been no flow rate of zero in the last year*

D.3.2.2.5.24 PreviousDayNConsumptionDelivered Attribute

PreviousDayNConsumptionDelivered represents the summed value of Energy, Gas, or Water delivered to the premises within the previous 24 hour period starting at the *Historical Freeze Time* (HFT). If the optional HFT attribute is not available, default to midnight local time.

D.3.2.2.5.25 PreviousDayNConsumptionReceived Attribute

PreviousDayNConsumptionReceived represents the summed value of Energy, Gas, or Water received from the premises within the previous 24 hour period starting at the *Historical Freeze Time* (HFT). If the optional HFT attribute is not available, default to midnight local time.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

6133 D.3.2.2.5.26 CurrentWeekConsumptionDelivered Attribute

6134 *CurrentWeekConsumptionDelivered* represents the summed value of Energy, Gas, or Water
6135 delivered to the premises since the *Historical Freeze Time* (HFT) on Monday to the last HFT
6136 read. If optionally provided, *CurrentWeekConsumptionDelivered* is updated continuously as new
6137 measurements are made. If the optional HFT attribute is not available, default to midnight local
6138 time.

6139 D.3.2.2.5.27 CurrentWeekConsumptionReceived Attribute

6140 *CurrentWeekConsumptionReceived* represents the summed value of Energy, Gas, or Water
6141 received from the premises since the *Historical Freeze Time* (HFT) on Monday to the last HFT
6142 read. If optionally provided, *CurrentWeekConsumptionReceived* is updated continuously as new
6143 measurements are made. If the optional HFT attribute is not available, default to midnight local
6144 time.

6145 D.3.2.2.5.28 PreviousWeekNConsumptionDelivered Attribute

6146 *PreviousWeekNConsumptionDelivered* represents the summed value of Energy, Gas, or Water
6147 delivered to the premises within the previous week period starting at the *Historical Freeze Time*
6148 (HFT) on the Monday to the Sunday. If the optional HFT attribute is not available, default to
6149 midnight local time.

6150 D.3.2.2.5.29 PreviousWeekNConsumptionReceived Attribute

6151 *PreviousWeekNConsumptionReceived* represents the summed value of Energy, Gas, or Water
6152 received from the premises within the previous week period starting at the *Historical Freeze*
6153 *Time* (HFT) on the Monday to the Sunday. If the optional HFT attribute is not available, default
6154 to midnight local time.

6155 D.3.2.2.5.30 CurrentMonthConsumptionDelivered Attribute

6156 *CurrentMonthConsumptionDelivered* represents the summed value of Energy, Gas, or Water
6157 delivered to the premises since the *Historical Freeze Time* (HFT) on the 1st of the month to the
6158 last HFT read. If optionally provided, *CurrentMonthConsumptionDelivered* is updated
6159 continuously as new measurements are made. If the optional HFT attribute is not available,
6160 default to midnight local time.

6161 D.3.2.2.5.31 CurrentMonthConsumptionReceived Attribute

6162 *CurrentMonthConsumptionReceived* represents the summed value of Energy, Gas, or Water
6163 received from the premises since the *Historical Freeze Time* (HFT) on the 1st of the month to the
6164 last HFT read. If optionally provided, *CurrentMonthConsumptionReceived* is updated
6165 continuously as new measurements are made. If the optional HFT attribute is not available,
6166 default to midnight local time.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

6167 D.3.2.2.5.32 PreviousMonthNConsumptionDelivered Attribute

6168 *PreviousMonthNConsumptionDelivered* represents the summed value of Energy, Gas, or Water
6169 delivered to the premises within the previous Month period starting at the *Historical Freeze Time*
6170 (HFT) on the 1st of the month to the last day of the month. If the optional HFT attribute is not
6171 available, default to midnight local time.

6172 D.3.2.2.5.33 PreviousMonthNConsumptionReceived Attribute

6173 *PreviousMonthNConsumptionReceived* represents the summed value of Energy, Gas, or Water
6174 received from the premises within the previous month period starting at the *Historical Freeze*
6175 *Time* (HFT) on the 1st of the month to the last day of the month. If the optional HFT attribute is
6176 not available, default to midnight local time.

6177 D.3.2.2.5.34 HistoricalFreezeTime Attribute

6178 *HistoricalFreezeTime* (HFT) represents the time of day, in Local Time, when Historical
6179 Consumption attributes and/or Alternative Historical Consumption attributes are captured.
6180 *HistoricalFreezeTime* is an unsigned 16-bit value representing the hour and minutes for HFT.
6181 The byte usages are:

6182 **Bits 0 to 7:** Range of 0 to 0x3B representing the number of minutes past the top of the hour.
6183
6184 **Bits 8 to 15:** Range of 0 to 0x17 representing the hour of the day (in 24-hour format). *Note*
6185 *that midnight shall be represented as 00:00 only*.

6186

6187 D.3.2.2.6 **Load Profile Configuration**

6188 The Load Profile Configuration Attribute Set is defined in Table D-30.

6189 **Table D-30– Load Profile Configuration Attribute Set**

Identifier	Name	Type	Range	Access	Default	Man./ Opt.
0x0500	*MaxNumberOfPeriods Delivered*	Unsigned 8 bit Integer	0x00 to 0xFF	Read Only	0x18	O
0x0501 to 0x05FF	Reserved					

6190

6191 D.3.2.2.6.1 MaxNumberOfPeriodsDelivered Attribute

6192 *MaxNumberofPeriodsDelivered* represents the maximum number of intervals the device is
6193 capable of returning in one *Get Profile Response* command. It is required
6194 *MaxNumberofPeriodsDelivered* fit within the default Fragmentation ASDU size of 128 bytes,

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

or an optionally agreed upon larger Fragmentation ASDU size supported by both devices. Please refer to sub-clause 5.3.8 for further details on Fragmentation settings.

D.3.2.2.7 **Supply Limit Attributes**

This set of attributes is used to implement a "Supply Capacity Limit" program where the demand at the premises is limited to a preset consumption level over a preset period of time. Should this preset limit be exceeded the meter could interrupt supply to the premises or to devices within the premises. The supply limit information in this attribute set can be used by In-Home displays, PCTs, or other devices to display a warning when the supply limit is being approached. The Supply Limit Attribute Set is defined in Table D-31.

Table D-31– Supply Limit Attribute Set

Identifier	Name	Type	Range	Access	Default	Man / Opt
0x0600	*CurrentDemand Delivered*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read only		O
0x0601	*DemandLimit*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read only		O
0x0602	*DemandIntegration Period*	Unsigned 8-bit Integer	0x01 to 0xFF	Read only	-	O
0x0603	*NumberOfDemand Subintervals*	Unsigned 8-bit Integer	0x01 to 0xFF	Read only	-	O
0x0604	DemandLimitArm Duration	Unsigned 16-bit Integer	0x0000 to 0xFFFF	Read only	0x003C	O
0x0605	LoadLimitSupplyState	8-bit Enumeration	0x00 to 0xFF	Read only	0x00	O
0x0606	LoadLimitCounter	Unsigned 8-bit Integer	0x00 to 0xFF	Read only	0x01	O
0x0607	SupplyTamperState	8-bit Enumeration	0x00 to 0xFF	Read only	0x00	O
0x0608	SupplyDepletionState	8-bit Enumeration	0x00 to 0xFF	Read only	0x00	O
0x0609	SupplyUncontrolled FlowState	8-bit Enumeration	0x00 to 0xFF	Read only	0x00	O
0x060A - 0x06FF	Reserved					

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

D.3.2.2.7.1 CurrentDemandDelivered Attribute

CurrentDemandDelivered represents the current Demand of Energy, Gas, or Water delivered at the premises. *CurrentDemandDelivered* may be continuously updated as new measurements are acquired, but at a minimum *CurrentDemandDelivered* must be updated at the end of each integration sub- period, which can be obtained by dividing the *DemandIntegrationPeriod* by the *NumberOfDemandSubintervals*.

This attribute shall be adjusted using the *Multiplier* and *Divisor* attributes found in the Formatting Attribute Set and can be formatted using the *DemandFormatting* attribute. The final result represents an engineering value in the unit defined by the *UnitofMeasure* attribute.

D.3.2.2.7.2 DemandLimit Attribute

DemandLimit reflects the current supply demand limit set in the meter. This value can be compared to the *CurrentDemandDelivered* attribute to understand if limits are being approached or exceeded.

Adjustment and formatting of this attribute follow the same rules as the *CurrentDemandDelivered*.

A value of “0xFFFFFF” indicates “demand limiting” is switched off.

D.3.2.2.7.3 DemandIntegrationPeriod Attribute

DemandIntegrationPeriod is the number of minutes over which the *CurrentDemandDelivered* attribute is calculated. Valid range is 0x01 to 0xFF. 0x00 is a reserved value.

D.3.2.2.7.4 NumberOfDemandSubintervals Attribute

NumberOfDemandSubintervals represents the number of subintervals used within the *DemandIntegrationPeriod*. The subinterval duration (in minutes) is obtained by dividing the *DemandIntegrationPeriod* by the *NumberOfDemandSubintervals*. The *CurrentDemandDelivered* attribute is updated at the each of each subinterval. Valid range is 0x01 to 0xFF. 0x00 is a reserved value.

As a Rolling Demand example, *DemandIntegrationPeriod* could be set at 30 (for 30 minute period) and *NumberOfDemandSubintervals* could be set for 6. This would provide 5 minute (30/6 = 5) subinterval periods.

As a Block Demand example, *DemandIntegrationPeriod* could be set at 30 (for 30 minute period) and *NumberOfDemandSubintervals* could be set for 1. This would provide a single 30 minute subinterval period.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

6238 D.3.2.2.7.5 DemandLimitArmDuration Attribute

6239 An unsigned 16-bit integer that defines the length of time, in seconds, that the supply shall be
6240 disconnected if the *DemandLimit* attribute is enabled and the limit is exceeded. At the end of the
6241 time period the meter shall move to the *ARMED* status. This will allow the user to reconnect the
6242 supply.

6243 D.3.2.2.7.6 LoadLimitSupplyState Attribute

6244 The *LoadLimitSupplyState* attribute indicates the required status of the supply once device is in a
6245 load limit state. The enumerated values for this field are outlined in Table D-68..

6246 D.3.2.2.7.7 LoadLimitCounter Attribute

6247 An unsigned 8-bit integer used for counting the number of times that the demand limit has
6248 exceeded the set threshold.

6249 This attribute shall be reset to zero on receipt of a *ResetLoadLimitCounter* command (see
6250 D.3.3.3.1.11 for further details).

6251 D.3.2.2.7.8 SupplyTamperState Attribute

6252 The *SupplyTamperState* indicates the required status of the supply following the detection of a
6253 tamper event within the metering device. The enumerated values for this field are outlined in
6254 Table D-68.

6255 D.3.2.2.7.9 SupplyDepletionState Attribute

6256 The *SupplyDepletionState* indicates the required status of the supply following detection of a
6257 depleted battery within the metering device. The enumerated values for this field are outlined in
6258 Table D-68.

6259 D.3.2.2.7.10 SupplyUncontrolledFlowState Attribute

6260 The *SupplyUncontrolledFlowState* indicates the required status of the supply following detection
6261 of an uncontrolled flow event within the metering device. The enumerated values for this field
6262 are outlined in Table D-68.

6263

6264 D.3.2.2.8 **Block Information Attribute Set (Delivered)**

6265 The following set of attributes provides a remote access to the Electric, Gas, or Water metering
6266 device's block readings. The Block Information attribute set supports Block pricing and
6267 combined Tier-Block pricing, the number of blocks is one greater than the number of block
6268 thresholds defined in the Pricing cluster.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

6269 This attribute set is ONLY for Energy, Gas or Water delivered to and consumed within the
6270 premises.

6271 **Table D-32– Block Information Attribute Set (Delivered)**

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0700	*CurrentNoTier Block1Summation Delivered*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0701	*CurrentNoTier Block2Summation Delivered*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0702	*CurrentNoTier Block3Summation Delivered*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x070N	... *CurrentNoTier BlockN+1 SummationDelivered*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x070F	*CurrentNoTier Block16Summation Delivered*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0710	*CurrentTier1 Block1Summation Delivered*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0711	*CurrentTier1 Block2Summation Delivered*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0712	*CurrentTier1 Block3Summation Delivered*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x071N	*CurrentTier1 BlockN+1 SummationDelivered* ...	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x071F	*CurrentTier1 Block16Summation Delivered*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0720	*CurrentTier2 Block1Summation Delivered*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x072N	*CurrentTier2 BlockN+1 SummationDelivered ...*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x072F	*CurrentTier2 Block16Summation Delivered*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0730	*CurrentTier3 Block1Summation Delivered*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x073N	*CurrentTier3 BlockN+1 SummationDelivered ...*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x073F	*CurrentTier3 Block16Summation Delivered*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0740	*CurrentTier4 Block1Summation Delivered*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x074N	*CurrentTier4 BlockN+1 SummationDelivered ...*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x074F	*CurrentTier4 Block16 SummationDelivered*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0750	*CurrentTier5 Block1Summation Delivered*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x075N	*CurrentTier5 BlockN+1 SummationDelivered ...*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x075F	*CurrentTier5 Block16Summation Delivered*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0760	*CurrentTier6 Block1Summation Delivered*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x076N	*CurrentTier6 BlockN+1 SummationDelivered ...*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x076F	*CurrentTier6 Block16Summation Delivered*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0770	*CurrentTier7 Block1Summation Delivered*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x077N	*CurrentTier7 BlockN+1 SummationDelivered* ...	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x077F	*CurrentTier7 Block16Summation Delivered*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0780	*CurrentTier8 Block1Summation Delivered*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x078N	*CurrentTier8 BlockN+1 SummationDelivered*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x078F	*CurrentTier8 Block16Summation Delivered*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0790	*CurrentTier9 Block1Summation Delivered*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x079N	*CurrentTier9 BlockN+1 SummationDelivered* ...	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x079F	*CurrentTier9 Block16Summation Delivered*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x07A0	*CurrentTier10 Block1Summation Delivered*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x07AN	*CurrentTier10 BlockN+1 SummationDelivered* ...	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x07AF	*CurrentTier10 Block16Summation Delivered*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x07B0	*CurrentTier11 Block1Summation Delivered*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x07BN	*CurrentTier11 BlockN+1 SummationDelivered ...*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x07BF	*CurrentTier11 Block16Summation Delivered*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x07C0	*CurrentTier12 Block1Summation Delivered*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x07CN	*CurrentTier12 BlockN+1 SummationDelivered ...*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x07CF	*CurrentTier12 Block16Summation Delivered*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x07D0	*CurrentTier13 Block1Summation Delivered*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x07DN	*CurrentTier13 BlockN+1 SummationDelivered ...*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x07DF	*CurrentTier13 Block16Summation Delivered*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x07E0	*CurrentTier14 Block1Summation Delivered*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x07EN	*CurrentTier14 BlockN+1 SummationDelivered ...*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x07EF	*CurrentTier14 Block16Summation Delivered*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x07F0	*CurrentTier15 Block1Summation Delivered*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x07FN	*CurrenTier15 BlockN+1 SummationDelivered ...*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x07FF	*CurrentTier15 Block16Summation Delivered*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O

6272

6273 D.3.2.2.8.1 CurrentTierNBlockNSummationDelivered Attributes

6274 Attributes *CurrentNoTierBlock1SummationDelivered* through
6275 *CurrentTier15Block16SummationDelivered* represent the most recent summed value of Energy,
6276 Gas, or Water delivered to the premises (i.e delivered to the customer from the utility) at a
6277 specific price tier as defined by a TOU schedule, Block Threshold or a real time pricing period.
6278 If optionally provided, attributes *CurrentNoTierBlock1SummationDelivered* through
6279 *CurrentTier15Block16SummationDelivered* are updated continuously as new measurements are
6280 made.

6281 ***Note:*** *SummationFormatting shall be used against the Block Information attribute set. The*
6282 *expected practical limit for the number of Block attributes supported is 64. The*
6283 *CurrentTierNBlockNSummationDelivered attributes are reset at the start of each Block*
6284 *Threshold Period.*

6285 D.3.2.2.9 **Alarms Set**

6286 The following set of attributes provides a means to control which alarms may be generated from
6287 the meter.

6288
6289 **Table D-33– Alarm Attribute Set**

Identifier	Name	Type	Range	Access	Default	Mandatoy / Optional
0x0800	Generic AlarmMask	16-bit BitMap	0x0000 - 0xffff	Read/ Write	0xffff	O
0x0801	Electricity AlarmMask	32-bit BitMap	0x00000000 - 0xffffffff	Read/ Write	0xffffffff	O
0x0802	Generic Flow/ Pressure AlarmMask	16-bit BitMap	0x0000 - 0xffff	Read/ Write	0xffff	O
0x0803	Water Specific AlarmMask	16-bit BitMap	0x0000 - 0xffff	Read/ Write	0xffff	O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x0804	Heat and Cooling Specific AlarmMask	16-bit BitMap	0x0000 - 0xffff	Read/ Write	0xffff	O
0x0805	Gas Specific AlarmMask	16-bit BitMap	0x0000 - 0xffff	Read/ Write	0xffff	O
0x0806	ExtendedGeneric AlarmMask	48-bit BitMap	0x000000000000 - 0xffffffffffff	Read/ Write	0xffffffffffff	O
0x0807	Manufacturer AlarmMask	16-bit BitMap	0x0000 - 0xffff	Read/ Write	0xffff	O
0x0808 – 0x08FF	Reserved					

6290

6291 D.3.2.2.9.1 AlarmMask Attributes

6292 The *AlarmMask* attributes of the Alarm Attribute Set specify whether each of the alarms listed in
6293 the corresponding alarm group in Table D-34 through Table D-42 is enabled. When the bit
6294 number corresponding to the alarm number (minus the group offset) is set to 1, the alarm is
6295 enabled, else it is disabled. Bits not corresponding to a code in the respective table are reserved.

6296 D.3.2.2.9.2 Alarm Codes

6297 The alarm codes are organized in logical groups corresponding to the meter type as listed below.
6298 The three main alarm groups are: Generic, Electricity, and Flow/ Pressure. The Flow/Pressure
6299 Alarm Group is further divided into Generic Flow/ Pressure, Water Specific, Heat and Cooling
6300 Specific, and Gas Specific. It is left for the manufacturer to select which (if any) alarm codes to
6301 support.

6302 **Table D-34– Alarm Code Groups**

Alarm Code	Alarm Condition
00-0F	**Generic Alarm Group**
10-2F	**Electricity Alarm Group**
30-6F	**Flow/Pressure Alarm Group** which is sub-divided as: **30-3F** - Generic Flow/Pressure Alarm Group **40-4F** - Water Specific Alarm Group **50-5F** - Heat and Cooling Specific Alarm Group **60-6F** - Gas Specific Alarm Group
70-AF	**Extended Generic Alarm Group**
B0-BF	**Manufacturer Specific Alarm Group**
C0-FF	**Reserved**

6303

6304 The generic Alarm Group maps the status from the *MeterStatus* attribute into a corresponding
6305 alarm. Hence, depending on the meter type, an alarm belonging to the Generic Alarm Group

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

may have a different meaning. See sub-clause D.3.2.2.3. In the case of overlap of alarm codes from the Generic Alarm Group with codes in other groups, e.g. Burst Detect, it is recommended to only use the code of the Generic Alarm Group.

Table D-35– Generic Alarm Group

Alarm Code	Alarm Condition
00	Check Meter
01	Low Battery
02	Tamper Detect
03	Electricity: Power Failure Gas: Not Defined Water: Pipe Empty Heat/Cooling: Temperature Sensor
04	Electricity: Power Quality Gas: Low Pressure Water: Low Pressure Heat/Cooling: Burst Detect
05	Leak Detect
06	Service Disconnect
07	Electricity: Reserved Gas: Reverse Flow Water: Reverse Flow Heat/Cooling: Flow Sensor
08	Meter Cover Removed
09	Meter Cover Closed
0A	Strong Magnetic Field
0B	No Strong Magnetic Field
0C	Battery Failure
0D	Program Memory Error
0E	RAM Error
0F	NV Memory Error

The Electricity Alarm Group defines alarms specific for electricity meters as defined below.

Table D-36– Electricity Alarm Group

Alarm Code	Alarm Condition
10	Low Voltage L1
11	High Voltage L1

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

12	Low Voltage L2
13	High Voltage L2
14	Low Voltage L3
15	High Voltage L3
16	Over Current L1
17	Over Current L2
18	Over Current L3
19	Frequency too Low L1
1A	Frequency too High L1
1B	Frequency too Low L2
1C	Frequency too High L2
1D	Frequency too Low L3
1E	Frequency too High L3
1F	Ground Fault
20	Electric Tamper Detect
21	Incorrect Polarity
22	Current No Voltage
23	Under Voltage
24	Over Voltage
25	Normal Voltage
26	PF Below Threshold
27	PF Above Threshold
28	Terminal Cover Removed
29	Terminal Cover Closed
2A-2F	Reserved

6313
6314 The Generic Flow/Pressure Alarm Group defines alarms specific for Flow/Pressure based meters
6315 i.e. Water, Heat, Cooling, or Gas meters as defined below.

6316 **Table D-37– Generic Flow/Pressure Alarm Group**

Alarm Code	Alarm Condition
30	Burst detect
31	Pressure too low
32	Pressure too high
33	Flow sensor communication error
34	Flow sensor measurement fault
35	Flow sensor reverse flow

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

36	Flow sensor air detect
37	Pipe empty
38-3F	Reserved

The Water Specific Alarm Group defines alarms specific for Water meters as defined below.

Table D-38– Water Specific Alarm Group

Alarm Code	Alarm Condition
40-4F	Reserved

The Heat and Cooling Specific Alarm Group defines alarms specific for Heat or Cooling meters as defined below.

Table D-39– Heat and Cooling Specific Alarm Group

Alarm Code	Alarm Condition
50	Inlet Temperature Sensor Fault
51	Outlet Temperature Sensor Fault
52-5F	Reserved

The Gas Specific Alarm Group defines alarms specific for Gas meters as defined below.

Table D-40– Gas Specific Alarm Group

Alarm Code	Alarm Condition
60	Tilt Tamper
61	Battery Cover Removed
62	Battery Cover Closed
63	Excess Flow
64	Tilt Tamper Ended
65-6F	Reserved

The Extended Generic Alarm Group is an additional set of generic meter alarms.

Table D-41– Extended Generic Alarm Group

Alarm Code	Alarm Condition
0x70	Measurement System Error
0x71	Watchdog Error
0x72	Supply Disconnect Failure
0x73	Supply Connect Failure
0x74	Measurment Software Changed

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Alarm Code	Alarm Condition
0x75	DST enabled
0x76	DST disabled
0x77	Clock Adj Backward (the internal clock has applied a negative adjustment)
0x78	Clock Adj Forward (the internal clock has applied a positive adjustment)
0x79	Clock Invalid
0x7A	Communication Error HAN
0x7B	Communication OK HAN
0x7C	Meter Fraud Attempt
0x7D	Power Loss
0x7E	Unusual HAN Traffic
0x7F	Unexpected Clock Change
0x80	Comms Using Unauthenticated Component
0x81	Error Reg Clear
0x82	Alarm Reg Clear
0x83	Unexpected HW Reset
0x84	Unexpected Program Execution
0x85	EventLog Cleared
0x86	Limit Threshold Exceeded
0x87	Limit Threshold OK
0x88	Limit Threshold Changed
0x89	Maximum Demand Exceeded
0x8A	Profile Cleared
0x8B	Sampling Buffer cleared
0x8C	Battery Warning
0x8D	Wrong Signature
0x8E	No Signature
0x8F	Unauthorised Action from HAN
0x90	Fast Polling Start
0x91	Fast Polling End
0x92	Meter Reporting Interval Changed
0x93	Disconnect Due to Load Limit
0x94	Meter Supply Status Register Changed
0x95	Meter Alarm Status Register Changed
0x96	Extended Meter Alarm Status Register Changed.
0x97 - AF	Reserved

6330
6331 The Manufacturer Specific Alarm Group defines alarms specific for any meters as defined
6332 below, these are used for meter specific functionality that is not covered by the current smart
6333 energy specification.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

6334 **Table D-42– Manufacturer Specific Alarm Group**

Alarm Code	Alarm Condition
0xB0	Manufacturer Specific A
0xB1	Manufacturer Specific B
0xB2	Manufacturer Specific C
0xB3	Manufacturer Specific D
0xB4	Manufacturer Specific E
0xB5	Manufacturer Specific F
0xB6	Manufacturer Specific G
0xB7	Manufacturer Specific H
0xB8	Manufacturer Specific I
0xB9 0xBF	Reserved
0xC0 – 0xC4	Reserved (command based events)
0xC5 – 0xFF	Reserved

6335

6336 D.3.2.2.10 **Block Information Attribute Set (Received)**

6337 The following set of attributes provides a remote access to the Electric, Gas, or Water metering
6338 devices block readings. The Block Information attribute set supports Block pricing and combined
6339 Tier-Block pricing, the number of blocks is one greater than the number of block thresholds
6340 defined in the Pricing cluster.

6341 This attribute set is ONLY for Energy generated from the premises and received by the utility.

6342 **Table D-43– Block Information Attribute Set (Received)**

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0900	*CurrentNoTier Block1Summation Received*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0901	*CurrentNoTier Block2Summation Received*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0902	*CurrentNoTier Block3Summation Received*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
…	...	…	…	…	…	…
0x090F	*CurrentNoTier Block16Summation Received*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x0910	*CurrentTier1 Block1Summation Received*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0911	*CurrentTier1 Block2Summation Received*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0912	*CurrentTier1 Block3Summation Received*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
…	...	…	…	…	…	…
0x091F	*CurrentTier1 Block16Summation Received*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0920	*CurrentTier2 Block1Summation Received*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
…	...	…	…	…	…	…
0x092F	*CurrentTier2 Block16Summation Received*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0930	*CurrentTier3 Block1Summation Received*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
…	...	…	…	…	…	…
0x093F	*CurrentTier3 Block16Summation Received*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0940	*CurrentTier4 Block1Summation Received*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
…	...	…	…	…	…	…
0x094F	*CurrentTier4 Block16 SummationReceived*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0950	*CurrentTier5 Block1Summation Received*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
…	...	…	…	…	…	…
0x095F	*CurrentTier5 Block16Summation Received*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x0960	*CurrentTier6 Block1Summation Received*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
…	...	…	…	…	…	…
0x096F	*CurrentTier6 Block16Summation Received*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0970	*CurrentTier7 Block1Summation Received*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
…	...	…	…	…	…	…
0x097F	*CurrentTier7 Block16Summation Received*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0980	*CurrentTier8 Block1Summation Received*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
…	...	…	…	…	…	…
0x098F	*CurrentTier8 Block16Summation Received*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0990	*CurrentTier9 Block1Summation Received*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
…	...	…	…	…	…	…
0x099F	*CurrentTier9 Block16Summation Received*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x09A0	*CurrentTier10 Block1Summation Received*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
…	...	…	…	…	…	…
0x09AF	*CurrentTier10 Block16Summation Received*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x09B0	*CurrentTier11 Block1Summation Received*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
…	...	…	…	…	…	…

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x09BF	*CurrentTier11 Block16Summation Received*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x09C0	*CurrentTier12 Block1Summation Received*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
…	…	…	…	…	…	…
0x09CF	*CurrentTier12 Block16Summation Received*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x09D0	*CurrentTier13 Block1Summation Received*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
…	…	…	…	…	…	…
0x09DF	*CurrentTier13 Block16Summation Received*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x09E0	*CurrentTier14 Block1Summation Received*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
…	…	…	…	…	…	…
0x09EF	*CurrentTier14 Block16Summation Received*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x09F0	*CurrentTier15 Block1Summation Received*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
…	…	…	…	…	…	…
0x09FF	*CurrentTier15 Block16Summation Received*	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O

D.3.2.2.10.1 CurrentTierNBlockNSummationReceived Attributes

Attributes *CurrentNoTierBlock1SummationReceived* through *CurrentTier15Block16SummationReceived* represent the most recent summed value of Energy, Gas, or Water received from the premises (i.e. delivered to the utility from the customer) at a specific price tier as defined by a TOU schedule, Block Threshold or a real time pricing period. If optionally provided, attributes *CurrentNoTierBlock1SummationReceived* through *CurrentTier15Block16SummationReceived* are updated continuously as new measurements are made.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Note: *SummationFormatting* shall be used against the Block Information attribute set. The practical limit for the number of Block attributes supported is 32. The *CurrentTierNBlockNSummationReceived* attributes are reset at the start of each Block Threshold Period.

D.3.2.2.11 **Meter Billing Attribute Set**

The billing information within this attribute set is created on the metering device. The information in this attribute set is intended for use by simple IHDs.

Table D-44– Meter Billing Attribute Set

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0A00	BillToDateDelivered	Unsigned 32 bit Integer	0x00000000 - 0xFFFFFFFF	Read Only	0x00	O
0x0A 01	BillToDateTimeStampDelivered	UTC Time		Read Only	0	O
0x0A 02	ProjectedBillDelivered	Unsigned 32 bit Integer	0x00000000 - 0xFFFFFFFF	Read Only	0x00	O
0x0A 03	ProjectedBillTimeStampDelivered	UTC Time		Read Only	0	O
0x0A04	BillDeliveredTrailingDigit	8-bit BitMap		Read Only		O
0x0A 05-0x0A 0F	Reserved					
0x0A 10	BillToDateReceived	Unsigned 32 bit Integer	0x00000000 - 0xFFFFFFFF	Read Only	0x00	O
0x0A 11	BillToDateTimeStampReceived	UTC Time		Read Only	0	O
0x0A 12	ProjectedBillReceived	Unsigned 32 bit Integer	0x00000000 - 0xFFFFFFFF	Read Only	0x00	O
0x0A 13	ProjectedBillTimeStampReceived	UTC Time		Read Only	0	O
0x0A14	BillReceivedTrailingDigit	8-bit BitMap		Read Only		O
0x0A 15 - 0x0A FF	Reserved					

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

6362 D.3.2.2.11.1 BillToDateDelivered Attribute

6363 *BillToDateDelivered* provides a value for the costs in the current billing period. This attribute is
6364 measured in a base unit of Currency with the decimal point located as indicated by the
6365 *BillDeliveredTrailingDigit* attribute.

6366 D.3.2.2.11.2 BillToDateTimeStampDelivered Attribute

6367 The UTC timestamp when the associated *BillToDateDelivered* attribute was last updated.

6368 D.3.2.2.11.3 *ProjectedBillDelivered* Attribute

6369 *ProjectedBillDelivered* provides a value indicating what the estimated state of the account will be
6370 at the end of the billing period based on past consumption. This attribute is measured in a base
6371 unit of Currency with the decimal point located as indicated by the *BillDeliveredTrailingDigit*
6372 attribute.

6373 D.3.2.2.11.4 *ProjectedBillTimeStampDelivered* Attribute

6374 The UTC timestamp when the associated *ProjectedBillDelivered* attribute was last updated.

6375 D.3.2.2.11.5 BillDeliveredTrailingDigit Attribute

6376 An 8-bit BitMap used to determine where the decimal point is located in the
6377 *BillToDateDelivered* and *ProjectedBillDelivered* attributes. The most significant nibble
6378 indicates the number of digits to the right of the decimal point. The least significant nibble is
6379 reserved and shall be 0. The *BillDeliveredTrailingDigit* attribute represents the current active
6380 value.

6381 D.3.2.2.11.6 *BillToDateReceived* Attribute

6382 *BillToDateReceived* provides a value for the costs in the current billing period. This attribute is
6383 measured in a base unit of Currency with the decimal point located as indicated by the
6384 *BillReceivedTrailingDigit* attribute.

6385 D.3.2.2.11.7 *BillToDateTimeStampReceived* Attribute

6386 The UTC timestamp when the associated *BillToDateReceived* attribute was last updated.

6387 D.3.2.2.11.8 *ProjectedBillReceived* Attribute

6388 *ProjectedBillReceived* provides a value indicating what the estimated state of the account will be
6389 at the end of the billing period based on past generation. This attribute is measured in a base unit
6390 of Currency with the decimal point located as indicated by the *BillReceivedTrailingDigit*
6391 attribute.

6392 D.3.2.2.11.9 *ProjectedBillTimeStampReceived* Attribute

6393 The UTC timestamp when the associated *ProjectedBillReceived* attribute was last updated.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

6394 D.3.2.2.11.10 BillReceivedTrailingDigit Attribute

6395 An 8-bit BitMap used to determine where the decimal point is located in the *BillToDateReceived*
6396 and *ProjectedBillReceived* attributes. The most significant nibble indicates the number of digits
6397 to the right of the decimal point. The least significant nibble is reserved and shall be 0. The
6398 *BillReceivedTrailingDigit* attribute represents the current active value.

6399

6400 D.3.2.2.12 **Supply Control Attribute Set**

6401 **Table D-45– Supply Control Attribute Set**

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0B00	ProposedChangeSupply ImplementationTime	UTC Time		Read only	-	O
0x0B01	ProposedChange SupplyStatus	8 bit Enumeration	0x00 to 0xFF	Read only	-	O
0x0B02 – 0x0B0F	Reserved					
0x0B10	Uncontrolled Flow Threshold	Unsigned 16-bit Integer		Read only	-	O
0x0B11	Uncontrolled Flow Threshold Unit of Measure	8 bit Enumeration		Read only	-	O
0x0B12	Uncontrolled Flow Multiplier	Unsigned 16-bit Integer		Read only	0x0001	O
0x0B13	Uncontrolled Flow Divisor	Unsigned 16-bit Integer		Read only	0x0001	O
0x0B14	Flow Stabilisation Period	Unsigned 8-bit Integer		Read only	-	O
0x0B15	Flow Measurement Period	Unsigned 16-bit Integer		Read only	-	O
0x0B16 – 0x0BFF	Reserved					

6402

6403 D.3.2.2.12.1 ProposedChangeSupplyImplementationTime Attribute

6404 The *ProposedChangeImplementationTime* attribute indicates the time at which a proposed
6405 change to the supply is to be implemented. If there is no change of supply pending, this attribute
6406 will be set to 0xFFFFFFFF.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

6407 D.3.2.2.12.2 ProposedChangeSupplyStatus Attribute

6408 The *ProposedChangeSupplyStatus* indicates the proposed status of the supply once the change to
6409 the supply has be been implemented. The enumerated values of this field are outlined in Table
6410 D-56

6411 D.3.2.2.12.3 Uncontrolled Flow Threshold Attribute

6412 The *Uncontrolled Flow Threshold* attribute indicates the threshold above which a flow meter
6413 (e.g. Gas or Water) shall detect an uncontrolled flow. A value of 0x0000 indicates the feature in
6414 unused.

6415 D.3.2.2.12.4 Uncontrolled Flow Threshold Unit of Measure Attribute

6416 The *Uncontrolled Flow Threshold Unit of Measure* attribute indicates the unit of measure used in
6417 conjunction with the *Uncontrolled Flow Threshold* attribute. The enumeration used for this field
6418 shall match one of the UnitOfMeasure values using a pure binary format as defined in this cluster
6419 (see sub-clause D.3.2.2.4.1).

6420 D.3.2.2.12.5 Uncontrolled Flow Multiplier Attribute

6421 The *Uncontrolled Flow Multiplier* attribute indicates the multiplier, to be used in conjunction
6422 with the *Uncontrolled Flow Threshold* and *Uncontrolled Flow Divisor* attributes, to determine
6423 the true flow threshold value. A value of 0x0000 is not allowed.

6424 D.3.2.2.12.6 Uncontrolled Flow Divisor Attribute

6425 The *Uncontrolled Flow Divisor* attribute indicates the divisor, to be used in conjunction with the
6426 *Uncontrolled Flow Threshold* and *Uncontrolled Flow Multiplier* attributes, to determine the true
6427 flow threshold value. A value of 0x0000 is not allowed.

6428 D.3.2.2.12.7 Flow Stabilisation Period Attribute

6429 The *Flow Stabilisation Period* attribute indicates the time given to allow the flow to stabilize. It
6430 is defined in units of tenths of a second.

6431 D.3.2.2.12.8 Flow Measurement Period Attribute

6432 The *Flow Measurement Period* attribute indicates the period over which the flow is measured
6433 and compared against the *Uncontrolled Flow Threshold* attribute. It is defined in units of 1
6434 second.
6435

6436 D.3.2.2.13 **Alternative Historical Consumption Attribute Set**

6437 The Alternative Historical Attribute Set allows historical information to be presented in a base
6438 defined by the *AlternativeUnitofMeasure* (see D.3.2.2.4.19) and in a format defined by the
6439 *AlternativeDemandFormatting* and *AlternativeConsumptionFormatting* attributes (see

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

6440 D.3.2.2.4.20 and D.3.2.2.4.21 respectively). The attributes within this set are defined in Table
6441 D-46.

6442 **Table D-46– Alternative Historical Consumption Attribute Set**

Identifier	Name	Type	Range	Access	Default	Man./ Opt.
0x0C00	*AlternativeInstantaneousDemand*	Signed 24-bit Integer	-8,388,607 to 8,388,607	Read Only	0x00	O
0x0C01	*CurrentDayAlternative ConsumptionDelivered*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0C02	*CurrentDayAlternative ConsumptionReceived*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0C03	*PreviousDayAlternative ConsumptionDelivered*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0C04	*PreviousDayAlternative ConsumptionReceived*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0C05	*CurrentAlternative PartialProfileInterval StartTimeDelivered*	UTCTime		Read Only	-	O
0x0C06	*CurrentAlternative PartialProfileInterval StartTimeReceived*	UTCTime	0x000000 to 0xFFFFFF	Read Only	-	O
0x0C07	*CurrentAlternative PartialProfileInterval ValueDelivered*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0C08	*CurrentAlternative PartialProfileInterval ValueReceived*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0C09	*CurrentDay AlternativeMaxPressure*	Unsigned 48-bit Integer	0x000000 000000 to 0xFFFFFF FFFFFF	Read Only	-	O
0x0C0A	*CurrentDay AlternativeMinPressure*	Unsigned 48-bit Integer	0x000000 000000 to 0xFFFFFF FFFFFF	Read Only	-	O
0x0C0B	*PreviousDay AlternativeMaxPressure*	Unsigned 48-bit Integer	0x000000 000000 to 0xFFFFFF FFFFFF	Read Only	-	O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x0C0C	*PreviousDay AlternativeMinPressure*	Unsigned 48-bit Integer	0x000000 000000 to 0xFFFFFF FFFFFF	Read Only	-	O
0x0C0D	*CurrentDay AlternativeMaxDemand*	Signed 24-bit Integer	-8,388,607 to 8,388,607	Read Only	-	O
0x0C0E	*PreviousDay AlternativeMaxDemand*	Signed 24-bit Integer	-8,388,607 to 8,388,607	Read Only	-	O
0x0C0F	*CurrentMonth AlternativeMaxDemand*	Signed 24-bit Integer	-8,388,607 to 8,388,607	Read Only	-	O
0x0C10	*CurrentYear AlternativeMaxDemand*	Signed 24-bit Integer	-8,388,607 to 8,388,607	Read Only	-	O
0x0C11 to 0x0C1F	Reserved					
0x0C20	*PreviousDay2Alternative ConsumptionDelivered*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0C21	*PreviousDay2Alternative ConsumptionReceived*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0C22	*PreviousDay3Alternative ConsumptionDelivered*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0C23	*PreviousDay3Alternative ConsumptionReceived*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0C24	*PreviousDay4Alternative ConsumptionDelivered*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0C25	*PreviousDay4Alternative ConsumptionReceived*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0C26	*PreviousDay5Alternative ConsumptionDelivered*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0C27	*PreviousDay5Alternative ConsumptionReceived*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0C28	*PreviousDay6Alternative ConsumptionDelivered*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0C29	*PreviousDay6Alternative ConsumptionReceived*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x0C2A	*PreviousDay7AlternativeConsumptionDelivered*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0C2B	*PreviousDay7AlternativeConsumptionReceived*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0C2C	*PreviousDay8AlternativeConsumptionDelivered*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0C2D	*PreviousDay8AlternativeConsumptionReceived*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0C2E to 0x0C2F	Reserved					
0x0C30	*CurrentWeekAlternativeConsumptionDelivered*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0C31	*CurrentWeekAlternativeConsumptionReceived*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0C32	*PreviousWeekAlternativeConsumptionDelivered*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0C33	*PreviousWeekAlternativeConsumptionReceived*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0C34	*PreviousWeek2AlternativeConsumptionDelivered*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0C35	*PreviousWeek2AlternativeConsumptionReceived*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0C36	*PreviousWeek3AlternativeConsumptionDelivered*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0C37	*PreviousWeek3AlternativeConsumptionReceived*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0C38	*PreviousWeek4AlternativeConsumptionDelivered*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0C39	*PreviousWeek4AlternativeConsumptionReceived*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x0C3A	*PreviousWeek5Alternative ConsumptionDelivered*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0C3B	*PreviousWeek5Alternative ConsumptionReceived*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0C3C to 0x0C3F	Reserved					
0x0C40	*CurrentMonthAlternative ConsumptionDelivered*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0C41	*CurrentMonthAlternative ConsumptionReceived*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0C42	*PreviousMonthAlternative ConsumptionDelivered*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0C43	*PreviousMonthAlternative ConsumptionReceived*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0C44	*PreviousMonth2Alternative ConsumptionDelivered*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0C45	*PreviousMonth2Alternative ConsumptionReceived*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0C46	*PreviousMonth3Alternative ConsumptionDelivered*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0C47	*PreviousMonth3Alternative ConsumptionReceived*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0C48	*PreviousMonth4Alternative ConsumptionDelivered*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0C49	*PreviousMonth4Alternative ConsumptionReceived*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0C4A	*PreviousMonth5Alternative ConsumptionDelivered*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0C4B	*PreviousMonth5Alternative ConsumptionReceived*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0C4C	*PreviousMonth6Alternative ConsumptionDelivered*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x0C4D	*PreviousMonth6Alternative ConsumptionReceived*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0C4E	*PreviousMonth7Alternative ConsumptionDelivered*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0C4F	*PreviousMonth7Alternative ConsumptionReceived*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0C50	*PreviousMonth8 Alternative ConsumptionDelivered*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0C51	*PreviousMonth8Alternative ConsumptionReceived*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0C52	*PreviousMonth9Alternative ConsumptionDelivered*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0C53	*PreviousMonth9Alternative ConsumptionReceived*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0C54	*PreviousMonth10Alternative ConsumptionDelivered*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0C55	*PreviousMonth10 Alternative ConsumptionReceived*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0C56	*PreviousMonth11 Alternative ConsumptionDelivered*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0C57	*PreviousMonth11Alternative ConsumptionReceived*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0C58	*PreviousMonth12Alternative ConsumptionDelivered*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0C59	*PreviousMonth12Alternative ConsumptionReceived*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0C5A	*PreviousMonth13Alternative ConsumptionDelivered*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0C5B	*PreviousMonth13Alternative ConsumptionReceived*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0C5C to 0x0CFF	Reserved					

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

6443

6444 D.3.2.2.13.1 AlternativeInstantaneousDemand Attribute

6445 *AlternativeInstantaneousDemand* represents the current Demand delivered or received at the
6446 premises. Positive values indicate demand delivered to the premises where negative values
6447 indicate demand received from the premises. *AlternativeInstantaneousDemand* is updated
6448 continuously as new measurements are made. The frequency of updates to this field is specific to
6449 the metering device, but should be within the range of once every second to once every 5 seconds.

6450 D.3.2.2.13.2 CurrentDayAlternativeConsumptionDelivered Attribute

6451 *CurrentDayAlternativeConsumptionDelivered* represents the summed value delivered to the
6452 premises since the *Historical Freeze Time* (HFT). If optionally provided,
6453 *CurrentDayAlternativeConsumptionDelivered* is updated continuously as new measurements are
6454 made. If the optional HFT attribute is not available, default to midnight local time.

6455 D.3.2.2.13.3 CurrentDayAlternativeConsumptionReceived Attribute

6456 *CurrentDayAlternativeConsumptionReceived* represents the summed value received from the
6457 premises since the *Historical Freeze Time* (HFT). If optionally provided,
6458 *CurrentDayAlternativeConsumptionReceived* is updated continuously as new measurements are
6459 made. If the optional HFT attribute is not available, default to midnight local time.

6460 D.3.2.2.13.4 PreviousDayAlternativeConsumptionDelivered Attribute

6461 *PreviousDayAlternativeConsumptionDelivered* represents the summed value delivered to the
6462 premises within the previous 24 hour period starting at the *lternative Historical Freeze Time*
6463 (HFT). If optionally provided, *PreviousDayAlternativeConsumptionDelivered* is updated every
6464 HFT. If the optional HFT attribute is not available, default to midnight local time.

6465 D.3.2.2.13.5 PreviousDayAlternativeConsumptionReceived Attribute

6466 *PreviousDayAlternativeConsumptionReceived* represents the summed value received from the
6467 premises within the previous 24 hour period starting at the *Historical Freeze Time* (HFT). If
6468 optionally provided, *PreviousDayAlternativeConsumptionReceived* is updated every HFT. If the
6469 optional HFT attribute is not available, default to midnight local time.

6470 D.3.2.2.13.6 CurrentAlternativePartialProfileIntervalStartTimeDelivered Attribute

6471 *CurrentAlternativePartialProfileIntervalStartTimeDelivered* represents the start time of the
6472 current Load Profile interval being accumulated for commodity delivered.

6473 D.3.2.2.13.7 CurrentAlternativePartialProfileIntervalStartTimeReceived Attribute

6474 *CurrentAlternativePartialProfileIntervalStartTimeReceived* represents the start time of the
6475 current Load Profile interval being accumulated for commodity received.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

D.3.2.2.13.8 CurrentAlternativePartialProfileIntervalValueDelivered Attribute

CurrentAlternativePartialProfileIntervalValueDelivered represents the value of the current Load Profile interval being accumulated for commodity delivered.

D.3.2.2.13.9 CurrentAlternativePartialProfileIntervalValueReceived Attribute

CurrentAlternativePartialProfileIntervalValueReceived represents the value of the current Load Profile interval being accumulated for commodity received.

D.3.2.2.13.10CurrentDayAlternativeMaxPressure Attribute

CurrentDayAlternativeMaxPressure is the maximum pressure reported during a day from the water or gas meter.

D.3.2.2.13.11PreviousDayAlternativeMaxPressure Attribute

PreviousDayAlternativeMaxPressure represents the maximum pressure reported during previous day from the water or gas meter.

D.3.2.2.13.12CurrentDayAlternativeMinPressure Attribute

CurrentDayAlternativeMinPressure is the minimum pressure reported during a day from the water or gas meter.

D.3.2.2.13.13PreviousDayAlternativeMinPressure Attribute

PreviousDayAlternativeMinPressure represents the minimum pressure reported during previous day from the water or gas meter.

D.3.2.2.13.14CurrentDayAlternativeMaxDemand Attribute

CurrentDayAlternativeMaxDemand represents the maximum demand or rate of delivered value of Energy, Gas, or Water being utilized at the premises.

D.3.2.2.13.15PreviousDayAlternativeMaxDemand Attribute

PreviousDayAlternativeMaxDemand represents the maximum demand or rate of delivered value of Energy, Gas, or Water being utilized at the premises.

Note: *At the end of a day the metering device will transfer the CurrentDayAlternativeMaxPressure into PreviousDayAlternativeMaxPressure, CurrentDayAlternativeMinPressure into PreviousDayAlternativeMinPressure and CurrentDayAlternativeMaxDemand into PreviousDayAlternativeMaxDemand.*

D.3.2.2.13.16CurrentMonthAlternativeMaxDemand Attribute

CurrentMonthAlternativeMaxDemand is the maximum demand reported during a month from the meter.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

For electricity, heat and cooling meters this is the maximum power reported in a month.

D.3.2.2.13.17 CurrentYearAlternativeMaxDemand Attribute

CurrentYearAlternativeMaxDemand is the maximum demand reported during a year from the meter.

For electricity, heat and cooling meters this is the maximum power reported in a year.

D.3.2.2.13.18 PreviousDayNAlternativeConsumptionDelivered Attribute

PreviousDayNAlternativeConsumptionDelivered represents the summed value delivered to the premises within the previous 24 hour period starting at the *Historical Freeze Time* (HFT). If the optional HFT attribute is not available, default to midnight local time.

D.3.2.2.13.19 PreviousDayNAlternativeConsumptionReceived Attribute

PreviousDayNAlternativeConsumptionReceived represents the summed value received from the premises within the previous 24 hour period starting at the *Historical Freeze Time* (HFT). If the optional HFT attribute is not available, default to midnight local time.

D.3.2.2.13.20 CurrentWeekAlternativeConsumptionDelivered Attribute

CurrentWeekAlternativeConsumptionDelivered represents the summed value delivered to the premises since the *Historical Freeze Time* (HFT) on Monday to the last HFT read. If optionally provided, *CurrentWeekAlternativeConsumptionDelivered* is updated continuously as new measurements are made. If the optional HFT attribute is not available, default to midnight local time.

D.3.2.2.13.21 CurrentWeekAlternativeConsumptionReceived Attribute

CurrentWeekAlternativeConsumptionReceived represents the summed value received from the premises since the *Historical Freeze Time* (HFT) on Monday to the last HFT read. If optionally provided, *CurrentWeekAlternativeConsumptionReceived* is updated continuously as new measurements are made. If the optional HFT attribute is not available, default to midnight local time.

D.3.2.2.13.22 PreviousWeekNAlternativeConsumptionDelivered Attribute

PreviousWeekNAlternativeConsumptionDelivered represents the summed value delivered to the premises within the previous week period starting at the *Historical Freeze Time* (HFT) on the Monday to the Sunday. If the optional HFT attribute is not available, default to midnight local time.

D.3.2.2.13.23 PreviousWeekNAlternativeConsumptionReceived Attribute

PreviousWeekNAlternativeConsumptionReceived represents the summed value received from the premises within the previous week period starting at the *Historical Freeze Time* (HFT) on the

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Monday to the Sunday. If the optional HFT attribute is not available, default to midnight local time.

D.3.2.2.13.24 CurrentMonthAlternativeConsumptionDelivered Attribute

CurrentMonthAlternativeConsumptionDelivered represents the summed value delivered to the premises since the *Historical Freeze Time* (HFT) on the 1st of the month to the last HFT read. If optionally provided, *CurrentMonthAlternativeConsumptionDelivered* is updated continuously as new measurements are made. If the optional HFT attribute is not available, default to midnight local time.

D.3.2.2.13.25 CurrentMonthAlternativeConsumptionReceived Attribute

CurrentMonthAlternativeConsumptionReceived represents the summed value received from the premises since the *Historical Freeze Time* (HFT) on the 1st of the month to the last HFT read. If optionally provided, *CurrentMonthAlternativeConsumptionReceived* is updated continuously as new measurements are made. If the optional HFT attribute is not available, default to midnight local time.

D.3.2.2.13.26 PreviousMonthNAlternativeConsumptionDelivered Attribute

PreviousMonthNAlternativeConsumptionDelivered represents the summed value delivered to the premises within the previous Month period starting at the *Historical Freeze Time* (HFT) on the 1st of the month to the last day of the month. If the optional HFT attribute is not available, default to midnight local time.

D.3.2.2.13.27 PreviousMonthNAlternativeConsumptionReceived Attribute

PreviousMonthNAlternativeConsumptionReceived represents the summed value received from the premises within the previous month period starting at the *Historical Freeze Time* (HFT) on the 1st of the month to the last day of the month. If the optional HFT attribute is not available, default to midnight local time.

D.3.2.3 Server Commands

D.3.2.3.1 **Commands Generated**

The command IDs generated by the Metering server cluster are listed in Table D-47.

Table D-47– Generated Command IDs for the Metering Server

Command Identifier Field Value	Description	Mandatory / Optional
0x00	*Get Profile Response*	O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x01	*Request Mirror*	O
0x02	*Remove Mirror*	O
0x03	*Request Fast Poll Mode Response*	O
0x04	*ScheduleSnapshot Response*	O
0x05	*TakeSnapshotResponse*	O
0x06	*Publish Snapshot*	O
0x07	*GetSampledData Response*	O
0x08	*ConfigureMirror*	O
0x09	*ConfigureNotification Scheme*	O
0x0A	*ConfigureNotification Flag*	O
0x0B	*GetNotifiedMessage*	O
0x0C	*Supply Status Response*	O
0x0D	*StartSamplingResponse*	O
0x0E – 0xff	Reserved	

6569

6570 D.3.2.3.1.1 Get Profile Response Command

6571 ***D.3.2.3.1.1.1 Payload Format***

6572 The *Get Profile Response* command payload shall be formatted as illustrated in Figure D-12.

Octets	**4**	**1**	**1**	**1**	**Variable**
Data Type	UTC Time	8-bit Enumeration	8-bit Enumeration	Unsigned 8-bit Integer	Series of Unsigned 24-bit Integers
Field Name	EndTime	Status	ProfileIntervalPeriod	NumberOfPeriods Delivered	Intervals

6573 **Figure D-12– *Get Profile Response* Command Payload**

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

6574 ***D.3.2.3.1.1.2 Payload Details***

6575 **EndTime:** 32-bit value (in UTC) representing the end time of the most chronologically
6576 recent interval being requested. Example: Data collected from 2:00 PM to 3:00 PM would be
6577 specified as a 3:00 PM interval (end time). It is important to note that the current interval
6578 accumulating is not included in most recent block but can be retrieved using the
6579 *CurrentPartialProfileIntervalValue* attribute.

6580 **Status:** Table D-48 lists the valid values returned in the Status field.

6581 **Table D-48– Status Field Values**

Value	Description
0x00	Success
0x01	Undefined Interval Channel requested
0x02	Interval Channel not supported
0x03	Invalid End Time
0x04	More periods requested than can be returned
0x05	No intervals available for the requested time
0x06 to 0xFF	Reserved for future use

6582

6583 **ProfileIntervalPeriod**: Represents the interval or time frame used to capture metered
6584 Energy, Gas, and Water consumption for profiling purposes. *ProfileIntervalPeriod* is
6585 an enumerated field representing the following timeframes listed in Table D-49:

6586 **Table D-49– ProfileIntervalPeriod Timeframes**

Enumerated Value	Timeframe
0	Daily
1	60 minutes
2	30 minutes
3	15 minutes
4	10 minutes
5	7.5 minutes
6	5 minutes
7	2.5 minutes
8 to 255	Reserved

6587

6588

6589 **NumberofPeriodsDelivered**: Represents the number of intervals the device is returning.
6590 Please note the number of periods returned in the *Get Profile Response* command can be

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

calculated when the packets are received and can replace the usage of this field. The intent is to provide this information as a convenience.

Intervals: Series of interval data captured using the period specified by the ProfileIntervalPeriod field. The content of the interval data depends of the type of information requested using the Channel field in the *Get Profile* Command, and will represent the change in that information since the previous interval. Data is organized in a reverse chronological order, the most recent interval is transmitted first and the oldest interval is transmitted last. Invalid intervals should be marked as 0xFFFFFF.

D.3.2.3.1.1.3 When Generated

This command is generated when the Client command *GetProfile* is received. Please refer to sub-clause D.3.3.3.1.1.

D.3.2.3.1.2 Request Mirror Command

This command is used to request the ESI to mirror Metering Device data.

D.3.2.3.1.2.1 Payload Details

There are no fields for this command.

D.3.2.3.1.2.2 Effect on Receipt

On receipt of this command, the Server shall send a *RequestMirrorReponse* command (see sub-clause D.3.3.3.1.2).

D.3.2.3.1.3 Remove Mirror Command

This command is used to request the ESI to remove its mirror of Metering Device data. The device sending the *Remove Mirror* command to the ESI shall send the command to the mirror endpoint to be removed. Only the device that created the mirror on the ESI or the ESI itself should be allowed to remove the mirror from the ESI.

D.3.2.3.1.3.1 Payload Details

There are no fields for this command.

D.3.2.3.1.3.2 Effect on Receipt

On receipt of this command, the Server shall send a *MirrorRemoved* command (see sub-clause D.3.3.3.1.3).

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

6622 D.3.2.3.1.4 Request Fast Poll Mode Response Command

6623 ***D.3.2.3.1.4.1 Payload Format***

6624 The *Request Fast Poll Mode Response* command payload shall be formatted as illustrated in
6625 Figure D-13:

Octets	1	4
Data Type	Unsigned 8-bit Integer	UTCTime
Field Name	Applied Update Period (seconds) (M)	Fast Poll Mode End Time (M)

6626 **Figure D-13– *Request Fast Poll Mode Response* Command Payload**

6627 ***D.3.2.3.1.4.2 Payload Details***

6628 **Applied Update Period:** The period at which metering data shall be updated. This may be
6629 different than the requested fast poll. If the Request Fast Poll Rate is less than *Fast Poll Update*
6630 *Period* Attribute, it shall use the Fast Poll Update Period Attribute. Otherwise, the Applied
6631 Update Period shall be greater than or equal to the minimum *Fast Poll Update Period* Attribute
6632 and less than or equal to the Requested Fast Poll Rate

6633 **Fast Poll Mode End Time:** UTC time that indicates when the metering server will terminate
6634 fast poll mode and resume updating at the rate specified by *DefaultUpdatePeriod*. For
6635 example, one or more metering clients may request fast poll mode while the metering server is
6636 already in fast poll mode. The intent is that the fast poll mode will not be extended since this
6637 scenario would make it possible to be in fast poll mode longer than 15 minutes.

6638 ***D.3.2.3.1.4.3 When Generated***

6639 This command is generated when the client command *Request Fast Poll Mode* is received.

6640 ***D.3.2.3.1.4.4 Effect on Receipt***

6641 On receipt of this command, the device may request or receive updates not to exceed the
6642 Applied Update Period until Fast Poll Mode End Time.

6643

6644 D.3.2.3.1.5 ScheduleSnapshotResponse Command

6645 This command is generated in response to a *ScheduleSnapshot* command, and is sent to confirm
6646 whether the requested snapshot schedule has been set up. See section D.3.4.5 for further details.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

6647 ***D.3.2.3.1.5.1 Payload Format***

Octets	4	Variable
Data Type	Unsigned 32 –bit Integer	
Field Name	Issuer Event ID (M)	Snapshot Response Payload (M)

6648 **Figure D-14– *ScheduleSnapshotResponse* Command Payload**

6649 ***D.3.2.3.1.5.2 Payload Details***

6650 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. The
6651 value contained in this field indicates the value allocated to the *ScheduleSnapshot* command for
6652 which this response is generated.

6653 ***D.3.2.3.1.5.3 Snapshot Response Payload***

6654 The *ScheduleSnapshotResponse* payload may contain several instances of the sub-payload
6655 defined in Figure D-15. Each instance is an acknowledgment from the device for a scheduled
6656 snapshot and the ability for the device to support that type of snapshot.

Octets	1	1
Data Type	Unsigned 8-bit Integer	Unsigned 8-bit Integer
Field Name	Snapshot Schedule ID (M)	Snapshot Schedule Confirmation (M)

6657 **Figure D-15– Snapshot Response Payload Sub-Payload**

6658

6659 **Snapshot Schedule ID (mandatory):** The unique ID of the Snapshot schedule; a range of 1-254
6660 is supported (see D.3.3.3.1.5.2 for further details).

6661 **Snapshot Schedule Confirmation (mandatory):** This provides confirmation for the Snapshot
6662 schedule; enumerations are defined in Table D-50.

6663 **Table D-50– Snapshot Schedule Confirmation**

Enumeration	Description
0x00	Accepted
0x01	Snapshot Type not supported
0x02	Snapshot Cause not supported
0x03	Snapshot Schedule Not Currently Available
0x04	Snapshot Schedules not supported by device
0x05	Insufficient space for snapshot schedule
0x06 – 0xFF	Reserved

6664

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

6665 D.3.2.3.1.6 TakeSnapshotResponse Command

6666 This command is generated in response to a *TakeSnapshot* command, and is sent to confirm
6667 whether the requested snapshot has been accepted and successfully taken. See section D.3.4.5 for
6668 further details.

6669 ***D.3.2.3.1.6.1 Payload Format***

Octets	**4**	**1**
Data Type	Unsigned 32 –bit Integer	Unsigned 8-bit Integer
Field Name	Snapshot ID (M)	Snapshot Confirmation (M)

6670 **Figure D-16– *TakeSnapshotResponse* Command Payload**

6671 ***D.3.2.3.1.6.2 Payload Details***

6672 **Snapshot ID (mandatory):** Unique identifier allocated by the device creating the snapshot. The
6673 value contained in this field indicates the *TakeSnapshot* command for which this response is
6674 generated.

6675 **Snapshot Confirmation (mandatory):** This is the acknowledgment from the device that it can
6676 support this required type of snapshot. The enumerations are defined in Table D-51.

6677 **Table D-51– Snapshot Confirmation**

Enumeration	**Description**
0x00	Accepted
0x01	Snapshot Cause not supported
0x02 – 0xFF	Reserved

6678

6679 D.3.2.3.1.7 Publish Snapshot Command

6680 This command is generated in response to a *GetSnapshot* command or when a new snapshot is
6681 created. It is used to return a single snapshot to the client. See section D.3.4.5 for further details.

6682 ***D.3.2.3.1.7.1 Payload Format***

Octets	**4**	**4**	**1**	**1**	**1**	**4**	**1**	**Variable**
Data Type	Unsigned 32 –bit Integer	UTC Time	Unsigned 8-bit Integer	Unsigned 8-bit Integer	Unsigned 8-bit Integer	32-bit BitMap	Unsigned 8-bit Enumeration	Snapshot type dependent
Field Name	Snapshot ID (M)	Snapshot Time (M)	Total Snapshots Found (M)	Command Index (M)	Total Number of Commands (M)	Snapshot Cause (M)	Snapshot Payload Type (M)	Snapshot Sub-Payload (M)

6683 **Figure D-17– *Publish Snapshot* Command Payload**

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

D.3.2.3.1.7.2 Payload Details

Snapshot ID (mandatory): Unique identifier allocated by the device creating the snapshot.

Snapshot Time (mandatory): This is a 32 bit value (in UTC Time) representing the time at which the data snapshot was taken.

Total Snapshots Found (mandatory): An 8-bit Integer indicating the number of snapshots found, based on the search criteria defined in the associated *GetSnapshot* command. If the value is greater than 1, the client is able to request the next snapshot by incrementing the *Snapshot Offset* field in an otherwise repeated *GetSnapshot* command.

Command Index (mandatory): The *CommandIndex* is used to count the payload fragments in the case where the entire payload (snapshot) does not fit into one message. The *CommandIndex* starts at 0 and is incremented for each fragment belonging to the same command.

Total Number of Commands (mandatory): In the case where the entire payload (snapshot) does not fit into one message, the *Total Number of Commands* field indicates the total number of sub-commands that will be returned.

Snapshot Cause (mandatory): A 32-bit BitMap indicating the cause of the snapshot. The snapshot cause values are listed in Table D-52.

Table D-52– Snapshot Cause BitMap

Bit	Cause Description
0	General
1	End of Billing Period
2	End of Block Period
3	Change of Tariff Information
4	Change of Price Matrix
5	Change of Block Thresholds
6	Change of CV
7	Change of CF
8	Change of Calendar
9	Critical Peak Pricing
10	Manually Triggered from Client
11	End of Resolve Period
12	Change of Tenancy
13	Change of Supplier
14	Change of (Meter) Mode
15	Debt Payment
16	Scheduled Snapshot
17	OTA Firmware Download
18	Reserved for Prepayment cluster

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Bit	Cause Description
19	Reserved for Prepayment cluster
20 – 31	Reserved

SnapshotPayloadType (mandatory): The *SnapshotPayloadType* is an 8-bit enumerator defining the format of the *SnapshotSub-Payload* in this message. The different snapshot types are listed in Table D-53. The server selects the *SnapshotPayloadType* based on the charging scheme in use.

Table D-53– Snapshot Payload Type

Enumeration	Description	Charging Scheme
0	TOU Information Set DeliveredRegisters	TOU charging only
1	TOU Information Set Received Registers	TOU charging only
2	Block Tier Information Set Delivered	Block/TOU charging
3	Block Tier Information Set Received	Block/TOU charging
4	TOU Information Set Delivered (No Billing)	TOU charging only
5	TOU Information Set Received (No Billing)	TOU charging only
6	Block Tier Information Set Delivered (No Billing)	Block/TOU charging
7	Block Tier Information Set Received (No Billing)	Block/TOU charging
8 - 127	Reserved	N/A
128	Data Unavailable	The data for this snapshot is currently unavailable; if used, there is currently no subsequent snapshot data.
129 - 255	Reserved	N/A

If the snapshot is taken by the server due to a change of Tariff Information (cause = 3) which involves a change in charging scheme then two snapshots shall be taken, the first according to the charging scheme being dismissed, the second to the scheme being introduced.

SnapshotSub-Payload (mandatory): the format of the *SnapshotSub-Payload* differs depending on the *SnapshotPayloadType*, as shown below. Note that, where the entire payload (snapshot) does not fit into one message, only the leading (non-Sub-Payload) fields of the Snapshot payload are repeated in each command; the *SnapshotSub-Payload* is divided over the required number of commands. Figure D-18 explains this further.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

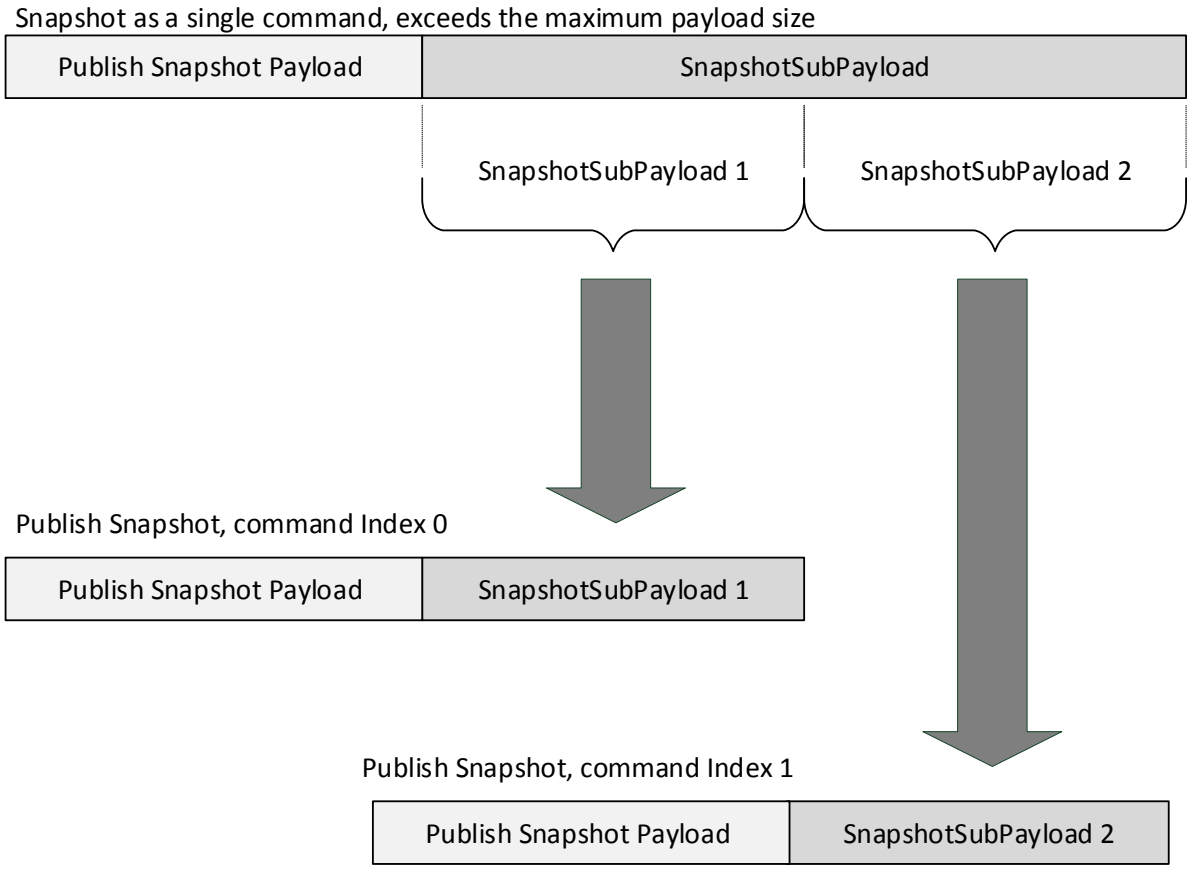


Figure D-18– Snapshot Utilizing Multiple Commands

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

a SnapshotPayloadType = TOU Information Delivered Set

6	4	4	4	4	1	1	Variable
Unsigned 48-bit Integer	Unsigned 32-bit Integer	UTC Time	Unsigned 32-bit Integer	UTC Time	8-bit BitMap	Unsigned 8-bit Integers	Series of Unsigned 48-bit Integers
Current Summation Delivered (M)	BillToDate Delivered (M)	BillToDate TimeStamp Delivered (M)	Projected Bill Delivered (M)	ProjectedBill TimeStamp Delivered (M)	Bill Delivered Trailing Digit (M)	Number of Tiers in Use (M)	Tier Summation (M)

Figure D-19– TOU Information Delivered Snapshot Sub-Payload

Current Summation Delivered (mandatory): An unsigned 48-bit integer that returns the value of the *CurrentSummationDelivered* attribute at the stated snapshot timestamp.

BillToDateDelivered (mandatory): An unsigned 32-bit integer that provides a value for the costs in the current billing period. This value is measured in a base unit of Currency with the decimal point located as indicated by the *BillDeliveredTrailingDigit* field.

BillToDateTimeStampDelivered (mandatory): A UTC timestamp that indicates when the value of the associated *BillToDateDelivered* parameter was last updated.

ProjectedBillDelivered (mandatory): An unsigned 32-bit integer that provides a value indicating what the estimated state of the account will be at the end of the billing period based on past consumption. This attribute is measured in a base unit of Currency with the decimal point located as indicated by the *BillDeliveredTrailingDigit* field.

ProjectedBillTimeStampDelivered (mandatory): A UTC timestamp that indicates when the associated *ProjectedBillDelivered* parameter was last updated.

BillDeliveredTrailingDigit (mandatory): An 8-bit BitMap used to determine where the decimal point is located in the *BillToDateDelivered* and *ProjectedBillDelivered* fields. The most significant nibble indicates the number of digits to the right of the decimal point. The least significant nibble is reserved and shall be 0.

Number of Tiers in Use (mandatory): An 8-bit integer representing the number of tiers in use at the time the snapshot was taken.

TierSummation (mandatory): The *Publish Snapshot* command contains N elements of *CurrentTierNSummationDelivered* attributes from the TOU Information Set. The Metering server shall send only the number of tiers in use, as stated in this command. The first element of the TOU Information Set (Delivered), is *CurrentTier1Summation*. For the following elements, the tier index is incremented until the number of tiers in use is reached.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

b SnapshotPayloadType = TOU Information Received Set

6	4	4	4	4	1	1	Variable
Unsigned 48-bit Integer	Unsigned 32-bit Integer	UTC Time	Unsigned 32-bit Integer	UTC Time	8-bit BitMap	Unsigned 8-bit Integers	Series of Unsigned 48-bit Integers
Current Summation Received (M)	BillToDate Received (M)	BillToDate TimeStamp Received (M)	Projected Bill Received (M)	ProjectedBill TimeStamp Received (M)	Bill Received Trailing Digit (M)	Number of Tiers in Use (M)	Tier Summation (M)

Figure D-20– TOU Information Received Snapshot Sub-Payload

Current Summation Received (mandatory): An unsigned 48-bit integer that returns the value of the *CurrentSummationReceived* attribute at the stated snapshot timestamp. A value of 0xFFFFFFFFFFFF means not available.

BillToDateReceived (mandatory): An unsigned 32-bit integer that provides a value for the costs in the current billing period. This value is measured in a base unit of Currency with the decimal point located as indicated by the *BillReceivedTrailingDigit* field.

BillToDateTimeStampReceived (mandatory): A UTC timestamp that indicates when the value of the associated *BillToDateReceived* parameter was last updated.

ProjectedBillReceived (mandatory): An unsigned 32-bit integer that provides a value indicating what the estimated state of the account will be at the end of the billing period based on past generation. This attribute is measured in a base unit of Currency with the decimal point located as indicated by the *BillReceivedTrailingDigit* field.

ProjectedBillTimeStampReceived (mandatory): A UTC timestamp that indicates when the associated *ProjectedBillReceived* parameter was last updated.

BillReceivedTrailingDigit (mandatory): An 8-bit BitMap used to determine where the decimal point is located in the *BillToDateReceived* and *ProjectedBillReceived* fields. The most significant nibble indicates the number of digits to the right of the decimal point. The least significant nibble is reserved and shall be 0.

Number of Tiers in Use (mandatory): An 8-bit integer representing the number of tiers in use at the time the snapshot was taken.

TierSummation (mandatory): The *Publish Snapshot* command contains N elements of *CurrentTierNSummationReceived* attributes from the TOU Information Set. The Metering server shall send only the number of tiers in use, as stated in this command. The first element

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

of the TOU Information Set (Received), is *CurrentTier1Summation*. For the following elements, the tier index is incremented until the number of tiers in use is reached.

c SnapshotPayloadType = Block Information Delivered Set

6	**4**	**4**	**4**	**4**	**1**
Unsigned 48-bit Integer	Unsigned 32-bit Integer	UTC Time	Unsigned 32-bit Integer	UTC Time	8-bit BitMap
Current Summation Delivered (M)	BillToDate Delivered (M)	BillToDate TimeStamp Delivered (M)	Projected Bill Delivered (M)	ProjectedBill TimeStamp Delivered (M)	Bill Delivered Trailing Digit (M)

1	**Variable**	**1**	**Variable**
Unsigned 8-bit Integers	Series of Unsigned 48-bit Integers	Unsigned 8-bit BitMap	Series of Unsigned 48-bit Integers
Number of Tiers in Use (M)	Tier Summation (M)	Number of Tiers and Block Thresholds in Use (M)	Tier Block Summation (M)

Figure D-21– Block Information Delivered Snapshot Sub-Payload

Current Summation Delivered (mandatory): An unsigned 48-bit integer that returns the value of the *CurrentSummationDelivered* attribute at the stated snapshot timestamp.

BillToDateDelivered (mandatory): An unsigned 32-bit integer that provides a value for the costs in the current billing period. This value is measured in a base unit of Currency with the decimal point located as indicated by the *BillDeliveredTrailingDigit* field.

BillToDateTimeStampDelivered (mandatory): A UTC timestamp that indicates when the value of the associated *BillToDateDelivered* parameter was last updated.

ProjectedBillDelivered (mandatory): An unsigned 32-bit integer that provides a value indicating what the estimated state of the account will be at the end of the billing period based on past consumption. This attribute is measured in a base unit of Currency with the decimal point located as indicated by the *BillDeliveredTrailingDigit* field.

ProjectedBillTimeStampDelivered (mandatory): A UTC timestamp that indicates when the associated *ProjectedBillDelivered* parameter was last updated.

BillDeliveredTrailingDigit (mandatory): An 8-bit BitMap used to determine where the decimal point is located in the *BillToDateDelivered* and *ProjectedBillDelivered* fields. The most significant nibble indicates the number of digits to the right of the decimal point. The least significant nibble is reserved and shall be 0.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Number of Tiers in Use (mandatory): An 8-bit integer representing the number of tiers in use at the time the snapshot was taken.

TierSummation (mandatory): The *Publish Snapshot* command contains N elements of *CurrentTierNSummationDelivered* attributes from the TOU Information Set. The Metering server shall send only the number of tiers in use, as stated in this command. The first element of the TOU Information Set (Delivered), is *CurrentTier1Summation*. For the following elements, the tier index is incremented until the number of tiers in use is reached.

Number of Tiers and Block Thresholds in Use (mandatory): An 8-bit BitMap representing the number of tiers and block thresholds in use at the time the snapshot was taken. The most significant nibble defines the number of tiers in use, whereas the least significant nibble indicates the number of block thresholds in use.

TierBlockSummation (T,B): The *Publish Snapshot* command contains N elements of the Block Information Attribute Set (Delivered). The metering server shall send only the number of Tiers and Blocks in use as stated in this command. The Block Information Attribute Set has two dimensions, the row – tier index (T) and the block – column index (B).

The first element of the Tier Block Summation field is *CurrentTier1Block1SummationDelivered* attribute. For the following elements, the block index is incremented until the number of blocks in use is reached. Then the tier index is incremented and the block index starts at 1 again. This continues until the stated number of tiers in use is reached.

d SnapshotPayloadType = Block Information Received Set

6	**4**	**4**	**4**	**4**	**1**
Unsigned 48-bit Integer	Unsigned 32-bit Integer	UTC Time	Unsigned 32-bit Integer	UTC Time	8-bit BitMap
Current Summation Received (M)	BillToDate Received (M)	BillToDate TimeStamp Received (M)	Projected Bill Received (M)	ProjectedBill TimeStamp Received (M)	Bill Received Trailing Digit (M)

1	**Variable**	**1**	**Variable**
Unsigned 8-bit Integers	Series of Unsigned 48-bit Integers	Unsigned 8-bit BitMap	Series of Unsigned 48-bit Integers
Number of Tiers in Use (M)	Tier Summation (M)	Number of Tiers and Block Thresholds in Use (M)	Tier Block Summation (M)

Figure D-22– Block Information Received Snapshot Sub-Payload

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Current Summation Received (mandatory): An unsigned 48-bit integer that returns the value of the *CurrentSummationReceived* attribute at the stated snapshot timestamp. A value of 0xFFFFFFFFFFFF means not available.

BillToDateReceived (mandatory): An unsigned 32-bit integer that provides a value for the costs in the current billing period. This value is measured in a base unit of Currency with the decimal point located as indicated by the *BillReceivedTrailingDigit* field.

BillToDateTimeStampReceived (mandatory): A UTC timestamp that indicates when the value of the associated *BillToDateReceived* parameter was last updated.

ProjectedBillReceived (mandatory): An unsigned 32-bit integer that provides a value indicating what the estimated state of the account will be at the end of the billing period based on past generation. This attribute is measured in a base unit of Currency with the decimal point located as indicated by the *BillReceivedTrailingDigit* field.

ProjectedBillTimeStampReceived (mandatory): A UTC timestamp that indicates when the associated *ProjectedBillReceived* parameter was last updated.

BillReceivedTrailingDigit (mandatory): An 8-bit BitMap used to determine where the decimal point is located in the *BillToDateReceived* and *ProjectedBillReceived* fields. The most significant nibble indicates the number of digits to the right of the decimal point. The least significant nibble is reserved and shall be 0.

Number of Tiers in Use (mandatory): An 8-bit integer representing the number of tiers in use at the time the snapshot was taken.

TierSummation (mandatory): The *Publish Snapshot* command contains N elements of *CurrentTierNSummationReceived* attributes from the TOU Information Set. The Metering server shall send only the number of tiers in use, as stated in this command. The first element of the TOU Information Set (Received), is *CurrentTier1Summation*. For the following elements, the tier index is incremented until the number of tiers in use is reached.

Number of Tiers and Block Thresholds in Use (mandatory): An 8-bit BitMap representing the number of tiers and block thresholds in use at the time the snapshot was taken. The most significant nibble defines the number of tiers in use, whereas the least significant nibble indicates the number of block thresholds in use.

TierBlockSummation (T,B): The *Publish Snapshot* command contains N elements of the Block Information Attribute Set (Received). The metering server shall send only the number of Tiers and Blocks in use as stated in this command. The Block Information Attribute Set has two dimensions, the row – tier index (T) and the block – column index (B).

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

The first element of the Tier Block Summation field is *CurrentTier1Block1SummationReceived* attribute. For the following elements, the block index is incremented until the number of blocks in use is reached. Then the tier index is incremented and the block index starts at 1 again. This continues until the stated number of tiers in use is reached.

e SnapshotPayloadType = TOU Information Set Delivered (No Billing)

6	1	Variable
Unsigned 48-bit Integer	Unsigned 8-bit Integers	Series of Unsigned 48-bit Integers
Current Summation Delivered (M)	Number of Tiers in Use (M)	Tier Summation (M)

Figure D-23– TOU Information Delivered (No Billing) Snapshot Sub-Payload

Current Summation Delivered (mandatory): An unsigned 48-bit integer that returns the value of the *CurrentSummationDelivered* attribute at the stated snapshot timestamp.

Number of Tiers in Use (mandatory): An 8-bit integer representing the number of tiers in use at the time the snapshot was taken.

TierSummation (mandatory): The *Publish Snapshot* command contains N elements of *CurrentTierNSummationDelivered* attributes from the TOU Information Set. The Metering server shall send only the number of tiers in use, as stated in this command. The first element of the TOU Information Set (Delivered), is *CurrentTier1Summation*. For the following elements, the tier index is incremented until the number of tiers in use is reached.

f SnapshotPayloadType = TOU Information Set Received (No Billing)

6	1	Variable
Unsigned 48-bit Integer	Unsigned 8-bit Integers	Series of Unsigned 48-bit Integers
Current Summation Received (M)	Number of Tiers in Use (M)	Tier Summation (M)

Figure D-24– TOU Information Received (No Billing) Snapshot Sub-Payload

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Current Summation Received (mandatory): An unsigned 48-bit integer that returns the value of the *CurrentSummationReceived* attribute at the stated snapshot timestamp. A value of 0xFFFFFFFFFFFF means not available.

Number of Tiers in Use (mandatory): An 8-bit integer representing the number of tiers in use at the time the snapshot was taken.

TierSummation (mandatory): The *Publish Snapshot* command contains N elements of *CurrentTierNSummationReceived* attributes from the TOU Information Set. The Metering server shall send only the number of tiers in use, as stated in this command. The first element of the TOU Information Set (Received), is *CurrentTier1Summation*. For the following elements, the tier index is incremented until the number of tiers in use is reached.

g SnapshotPayloadType = Block Tier Information Set Delivered (No Billing)

6	1	Variable	1	Variable
Unsigned 48-bit Integer	Unsigned 8-bit Integers	Series of Unsigned 48-bit Integers	Unsigned 8-bit BitMap	Series of Unsigned 48-bit Integers
Current Summation Delivered (M)	Number of Tiers in Use (M)	Tier Summation (M)	Number of Tiers and Block Thresholds in Use (M)	Tier Block Summation (M)

Figure D-25– Block Information Delivered (No Billing) Snapshot Sub-Payload

Current Summation Delivered (mandatory): An unsigned 48-bit integer that returns the value of the *CurrentSummationDelivered* attribute at the stated snapshot timestamp.

Number of Tiers in Use (mandatory): An 8-bit integer representing the number of tiers in use at the time the snapshot was taken.

TierSummation (mandatory): The *Publish Snapshot* command contains N elements of *CurrentTierNSummationDelivered* attributes from the TOU Information Set. The Metering server shall send only the number of tiers in use, as stated in this command. The first element of the TOU Information Set (Delivered), is *CurrentTier1Summation*. For the following elements, the tier index is incremented until the number of tiers in use is reached.

Number of Tiers and Block Thresholds in Use (mandatory): An 8-bit BitMap representing the number of tiers and block thresholds in use at the time the snapshot was taken. The most significant nibble defines the number of tiers in use, whereas the least significant nibble indicates the number of block thresholds in use.

TierBlockSummation (T,B): The *Publish Snapshot* command contains N elements of the Block Information Attribute Set (Delivered). The metering server shall send only the number

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

of Tiers and Blocks in use as stated in this command. The Block Information Attribute Set has two dimensions, the row – tier index (T) and the block – column index (B).

The first element of the Tier Block Summation field is *CurrentTier1Block1SummationDelivered* attribute. For the following elements, the block index is incremented until the number of blocks in use is reached. Then the tier index is incremented and the block index starts at 1 again. This continues until the stated number of tiers in use is reached.

h SnapshotPayloadType = Block Tier Information Set Received (No Billing)

6	1	Variable	1	Variable
Unsigned 48-bit Integer	Unsigned 8-bit Integers	Series of Unsigned 48-bit Integers	Unsigned 8-bit BitMap	Series of Unsigned 48-bit Integers
Current Summation Received (M)	Number of Tiers in Use (M)	Tier Summation (M)	Number of Tiers and Block Thresholds in Use (M)	Tier Block Summation (M)

Figure D-26– Block Information Received (No Billing) Snapshot Sub-Payload

Current Summation Received (mandatory): An unsigned 48-bit integer that returns the value of the *CurrentSummationReceived* attribute at the stated snapshot timestamp. A value of 0xFFFFFFFFFFFF means not available.

Number of Tiers in Use (mandatory): An 8-bit integer representing the number of tiers in use at the time the snapshot was taken.

TierSummation (mandatory): The *Publish Snapshot* command contains N elements of *CurrentTierNSummationReceived* attributes from the TOU Information Set. The Metering server shall send only the number of tiers in use, as stated in this command. The first element of the TOU Information Set (Received), is *CurrentTier1Summation*. For the following elements, the tier index is incremented until the number of tiers in use is reached.

Number of Tiers and Block Thresholds in Use (mandatory): An 8-bit BitMap representing the number of tiers and block thresholds in use at the time the snapshot was taken. The most significant nibble defines the number of tiers in use, whereas the least significant nibble indicates the number of block thresholds in use.

TierBlockSummation (T,B): The *Publish Snapshot* command contains N elements of the Block Information Attribute Set (Received). The metering server shall send only the number of Tiers and Blocks in use as stated in this command. The Block Information Attribute Set has two dimensions, the row – tier index (T) and the block – column index (B).

The first element of the Tier Block Summation field is *CurrentTier1Block1SummationReceived* attribute. For the following elements, the block index

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

is incremented until the number of blocks in use is reached. Then the tier index is incremented and the block index starts at 1 again. This continues until the stated number of tiers in use is reached.

D.3.2.3.1.7.3 When Generated

A *Publish Snapshot* command is generated in response to *GetSnapshot* command or when a new snapshot is created. The device shall send a single *Publish Snapshot* command according to the search criteria defined in the associated *GetSnapshot* command. A ZCL Default Response with status NOT_FOUND shall be returned if there is no appropriate snapshot data available.

D.3.2.3.1.8 GetSampledDataResponse Command

This command is used to send the requested sample data to the client. It is generated in response to a *GetSampledData* command (see D.3.3.3.1.9).

D.3.2.3.1.8.1 Payload Format

Octets	**2**	**4**	**1**	**2**	**2**	**Variable**
Data Type	Unsigned 16 –bit Integer	UTC Time	8-bit Enumeration	Unsigned 16-bit Integer	Unsigned 16-bit Integer	Series of unsigned 24-bit Integers
Field Name	Sample ID (M)	SampleStart Time (M)	SampleType (M)	SampleRequest Interval (M)	NumberOfSamples (M)	Samples (M)

Figure D-27– Format of the *GetSampledDataResponse* Command Payload

D.3.2.3.1.8.2 Payload Details

SampleID (mandatory): Unique identifier allocated to this Sampling session. This field allows devices to match response data with the appropriate request. See D.3.2.3.1.14 for further details.

SampleStartTime (mandatory): A UTC Time field to denote the time of the first sample returned in this response.

SampleType (mandatory): An 8 bit enumeration that identifies the type of data being sampled. Possible values are defined in the following table:

Table D-54– Sample Type Enumerations

Enumeration	**Description**
0	Consumption Delivered
1 - 255	Reserved

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

SampleRequestInterval (mandatory): An unsigned 16-bit field representing the interval or time in seconds between samples.

NumberOfSamples (mandatory): Represents the number of samples being requested, This value cannot exceed the size stipulated in the MaxNumberofSamples field in the *StartSampling* command. If more samples are requested than can be delivered, the *GetSampleDataResponse* command will return the number of samples equal to MaxNumberofSamples field. If fewer samples are available for the time period, only those available shall be returned.

Samples (mandatory): Series of data samples captured using the interval specified by the *SampleRequestInterval* field in the *StartSampling* command. Each sample contains the change in the relevant data since the previous sample. Data is organised in a chronological order, the oldest sample is transmitted first and the most recent sample is transmitted last. Invalid samples should be marked as 0xFFFFFF.

D.3.2.3.1.8.3 When Generated

A *GetSampledDataResponse* command is generated in response to *GetSampledData* command. A ZCL Default Response with status NOT_FOUND shall be returned if there is no appropriate Sample data available.

D.3.2.3.1.9 ConfigureMirror Command

Where 'Two Way Mirroring' is being implemented, this command shall be sent to the mirror once the mirror has been created. The command allows a BOMD to provide the operational configuration of the associated Mirror. Note that this command is not required for a traditional 'One way' mirror (see D.3.4.4.3 for further details).

D.3.2.3.1.9.1 Payload Format

Octets	**4**	**3**	**1**	**1**
Data Type	Unsigned 32-bit Integer	Unsigned 24-bit Integer	Boolean	Unsigned 8-bit Integer
Field Name	Issuer Event ID (M)	Reporting Interval (M)	Mirror Notification Reporting (M)	Notification Scheme (M)

Figure D-28– Format of the *ConfigureMirror* Command Payload

D.3.2.3.1.9.2 Payload Details

Issuer Event ID (mandatory): Unique identifier generated by the device being mirrored. When new information is provided that replaces older information, this field allows devices to

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

determine which information is newer. It is recommended that the value contained in this field is a UTC based time stamp (UTCTime data type) identifying when the command was issued. Thus, newer information will have a value in the Issuer Event ID field that is larger than older information.

Reporting Interval (mandatory): An unsigned 24-bit integer to denote the interval, in seconds, at which a mirrored meter intends to use the *ReportAttribute* command..

Mirror Notification Reporting (mandatory): A Boolean used to advise a BOMD how the Notification flags should be acquired (see below).

When *Mirror Notification Reporting* is set, the *MirrorReportAttributeResponse* command is enabled. In that case, the Metering client on the mirror endpoint shall respond to the last or only *ReportAttribute* command with the *MirrorReportAttributeResponse*. This is shown if Figure D-29:

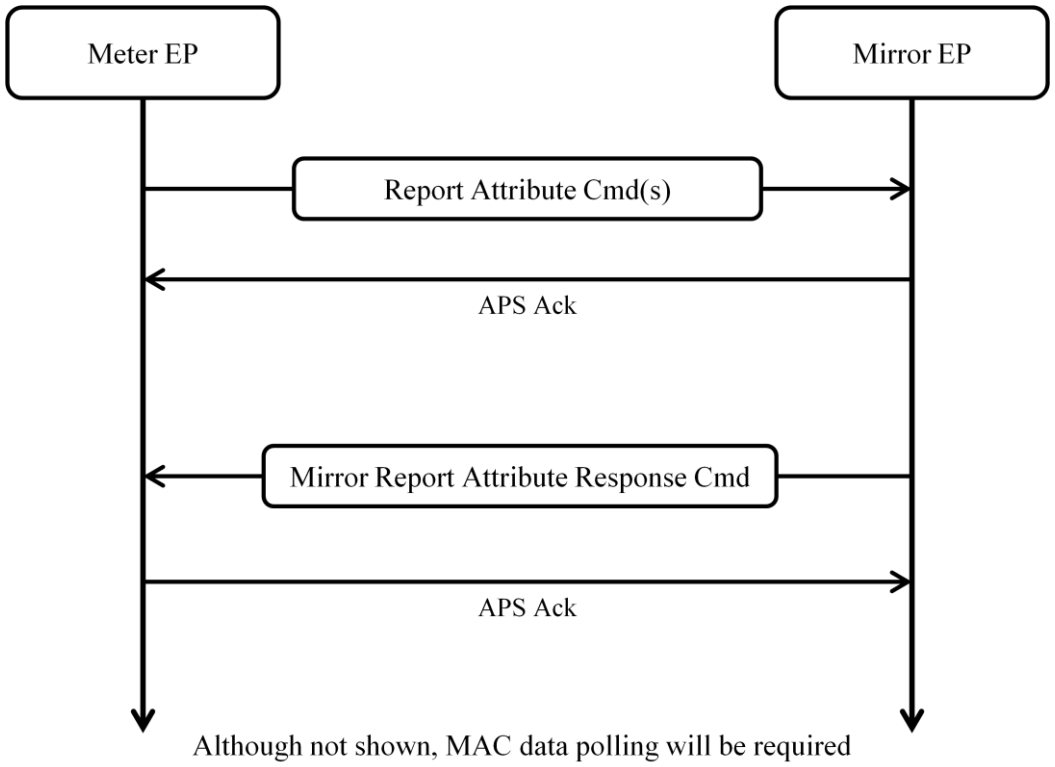


Figure D-29– *MirrorReportAttributeResponse* Command Enabled

NOTES:

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

1. On powering up, the BOMD will send one or more *Report Attribute* commands to the Metering client on the mirror endpoint. The last attribute to be reported to the mirror shall be an *Attribute Reporting Status* attribute, as defined in section A.2.

2. If *MirrorReportAttributeResponse* is enabled, the server does not need to request an APS ACK. If the server requests an APS ACK, the Metering client on the mirror endpoint shall respond first with an APS ACK and then send the *MirrorReportAttributeResponse*.

If *Mirror Notification Reporting* is set to FALSE, the *MirrorReportAttributeResponse* command shall not be enabled; the Metering server may poll the Notification flags by means of a normal *ReadAttribute* command, as shown in Figure D-30:

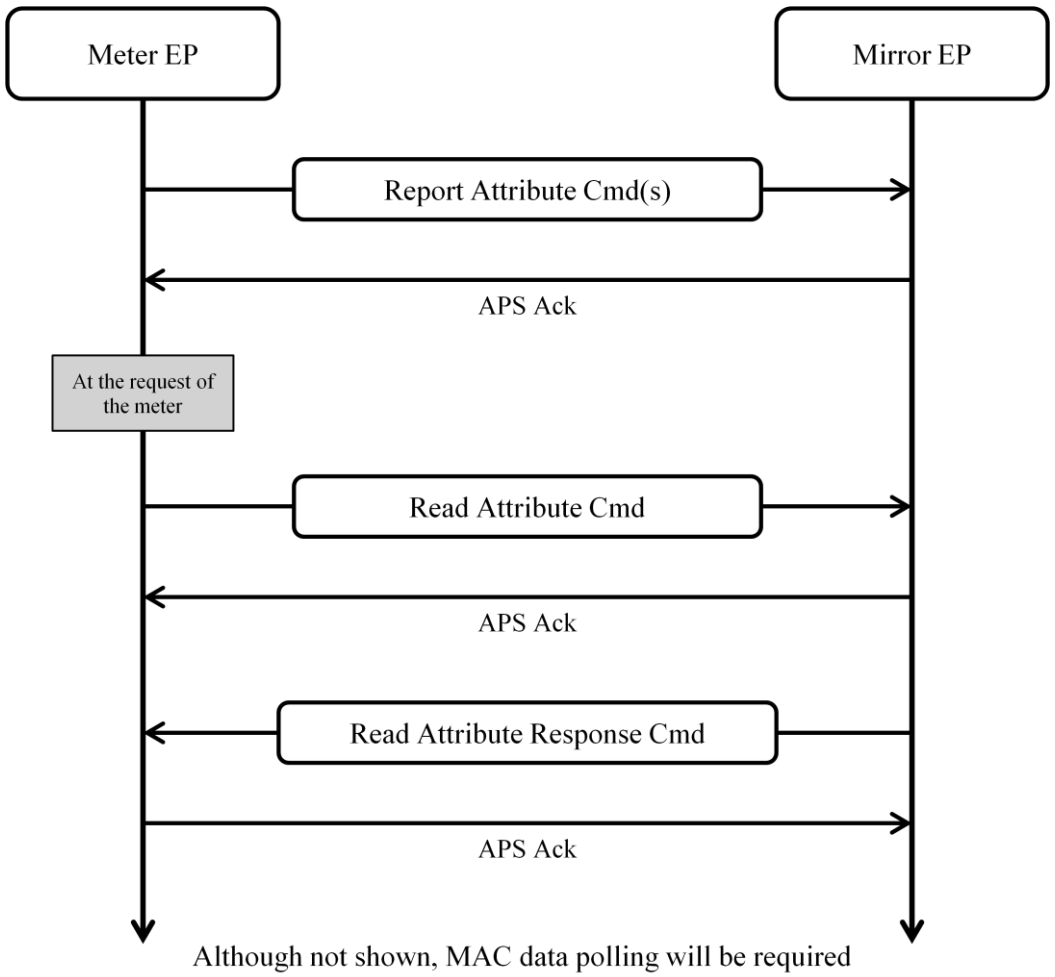


Figure D-30– *MirrorReportAttributeResponse* Command Disabled

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Notification Scheme (mandatory): This unsigned 8-bit integer allows for the pre-loading of the Notification Flags bit mapping to ZCL or Smart Energy Standard commands. The following schemes are currently supported within the Smart Energy Standard:-

Value	Description
0x00	No Notification Scheme Defined
0x01	Predefined Notification Scheme A
0x02	Predefined Notification Scheme B
0x03 – 0x80	Reserved
0x81 – 0xFE	For MSP Requirements
0xFF	Reserved

Figure D-31– *NotificationScheme* Enumerations

D.3.2.3.1.9.3 When Generated

The *ConfigureMirror* command is generated in response to the *RequestMirrorResponse* command when the Mirror has been created.

D.3.2.3.1.9.4 Effect on Receipt

On receipt of the *ConfigureMirror* command, the mirror will understand if the *MirrorReportAttributeResponse* command should be sent, and if there is a scheme for the Notifications flags. The Mirror will also understand the interval at which the Meter shall report to the mirror.

A ZCL Default Response with status INVALID_FIELD shall be returned if the required Notification Scheme is not supported by the Mirroring device.

D.3.2.3.1.10 ConfigureNotificationScheme Command

Note: *The ConfigureNotificationScheme command in this revision of this specification is provisional and not certifiable. This feature may change before reaching certifiable status in a future revision of this specification.*

Where 'Two Way Mirroring' is being implemented, and a non-default Notification Scheme is to be used, this command shall be sent to the mirror once the mirror has been created. The command allows a BOMD to provide details of the required Notification Scheme to the associated mirror, and should be used in conjunction with the associated *ConfigureNotificationFlags* command (see D.3.2.3.1.11). No default schemes are allowed to be overwritten (see D.3.4.4.3.4 and D.3.4.4.3.5 for further details); generic schemes should use one of the reserved values 0x03 – 0x80, MSP schemes should use one of the values 0x081-0x0FE (see Figure D-31). Section D.3.4.4.3.3 provides further details.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

7031 ***D.3.2.3.1.10.1 Payload Format***

Octets	4	1	4
Data Type	Unsigned 32-bit Integer	Unsigned 8-bit Integer	32-bit BitMap
Field Name	Issuer Event ID (M)	Notification Scheme (M)	Notification Flag Order (M)

7032 **Figure D-32– *ConfigureNotificationScheme* Command Payload**

7033 ***D.3.2.3.1.10.2 Payload Details***

7034 **Issuer Event ID (mandatory):** Unique identifier generated by the device being mirrored. When
7035 new information is provided that replaces older information, this field allows devices to
7036 determine which information is newer. It is recommended that the value contained in this field is
7037 a UTC based time stamp (UTCTime data type) identifying when the command was issued. Thus,
7038 newer information will have a value in the Issuer Event ID field that is larger than older
7039 information.

7040 **Notification Scheme (mandatory):** An unsigned 8-bit integer that allows for the pre-loading of
7041 the Notification Flags bit mapping to ZCL or Smart Energy Standard commands. Figure D-31
7042 details the schemes that are currently supported within the Smart Energy Standard.

7043 **Notification Flag Order (mandatory):** A 32-bit bitmap, consisting of 8 nibbles which define
7044 the Notification Flag attributes (and order) to be returned in a *MirrorReportAttributeResponse*
7045 command. The values to be returned in each nibble are defined in Table D-55.

7046 **Table D-55– Notification Flags Order**

Value	Waiting Command
0	NotificationFlag1
1	NotificationFlag2
2	NotificationFlag3
3	NotificationFlag4
4	NotificationFlag5
5	NotificationFlag6
6	NotificationFlag7
7	NotificationFlag8
8 –E	Reserved
F	Blank / No Notification Flag

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

7047 ***D.3.2.3.1.10.3 When Generated***

7048 The *ConfigureNotificationScheme* command is generated when a new scheme is required.

7049 ***D.3.2.3.1.10.4 Effect on Receipt***

7050 On receipt of the *ConfigureNotificationScheme* command, the mirror shall store the
7051 NotificationScheme information, and wait for the associated *ConfigureNotificationFlags*
7052 command. Until all of the *ConfigureNotificationFlags* commands have been received, the two-
7053 way mirror functionality should be disabled. The Notification Flag Order parameter will allow
7054 the mirror to determine when all of the *ConfigureNotificationFlags* commands have been
7055 received.

7056

7057 D.3.2.3.1.11 ConfigureNotificationFlags Command

7058 ***Note:*** *The ConfigureNotificationFlags command in this revision of this specification is*
7059 *provisional and not certifiable. This feature may change before reaching certifiable status in a*
7060 *future revision of this specification.*

7061 Where 'Two Way Mirroring' is being implemented, and a non-default Notification Scheme is to
7062 be used, the *ConfigureNotificationFlags* command allows a BOMD to set the commands relating
7063 to the bit value for each *NotificationFlags#N* attribute that the scheme is proposing to use. This
7064 command should be used in conjunction with the associated *ConfigureNotificationScheme*
7065 command (see D.3.2.3.1.10). No predefined schemes are allowed to be overwritten (see
7066 D.3.4.4.3.4 and D.3.4.4.3.5 for further details). Section D.3.4.4.3.3 provides further details.

7067 ***D.3.2.3.1.11.1 Payload Format***

Octets	**4**	**1**	**2**	**Variable**
Data Type	Unsigned 32-bit Integer	Unsigned 8-bit Integer	Unsigned 16-bit Integer	-
Field Name	Issuer Event ID (M)	Notification Scheme (M)	Notification Flag Attribute ID (M)	Bit Field Allocation

7068 **Figure D-33– *ConfigureNotificationFlags* Command Payload**

7069 ***D.3.2.3.1.11.2 Payload Details***

7070 **Issuer Event ID (mandatory):** Unique identifier generated by the device being mirrored. When
7071 new information is provided that replaces older information, this field allows devices to
7072 determine which information is newer. It is recommended that the value contained in this field is
7073 a UTC based time stamp (UTCTime data type) identifying when the command was issued. Thus,
7074 newer information will have a value in the Issuer Event ID field that is larger than older
7075 information.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Notification Scheme (mandatory): An unsigned 8-bit integer that allows for the pre-loading of the Notification Flags bit mapping to ZCL or Smart Energy Standard commands. Figure D-31 details the schemes that are currently supported within the Smart Energy Standard.

Notification Flag Attribute ID (mandatory): An unsigned 16-bit integer that denotes the attribute id of the Notification flag (2-8) that will be configured for this Notification scheme.

Bit Field Allocation (mandatory): The bit field allocation sub payload is defined in Figure D-34. The bit order is defined by the position of sub-payload within the command.

2	**2**	**1**	**1 to 32**		
Unsigned 16-bit Integer	Unsigned 16-bit Integer	Unsigned 8-bit Integer	Unsigned 8-bit Integer		Unsigned 8-bit Integer
Cluster ID	Manufacturer Code	No. of Commands	Command 1 Identifier	…	Command n Identifier

Figure D-34– *Bit Field Allocation* Command Sub Payload

Cluster ID (mandatory): An unsigned 16-bit integer that denotes the Cluster id of the Notification flag that will be configured for this Notification scheme.

Manufacturer Code (mandatory): An unsigned 16-bit integer that denotes the Manufacturer Code to be used with these command IDs, that are configured for this Notification flag within this Notification scheme.

No of Commands (mandatory): An unsigned 8-bit integer that indicates the number of command identifiers contained within this sub payload.

Command ID (mandatory): An unsigned 8-bit integer that denotes the command that is to be used. The command id should be used with the cluster id to reference the command(s).

D.3.2.3.1.11.3 When Generated

This command is sent once the mirror has been created, and the *ConfigureNotificationScheme* command has been sent up the top level of the scheme. There is a *ConfigureNotificationFlags* command for each attribute that the scheme is proposing to use. No default schemes are allowed to be overwritten.

D.3.2.3.1.11.4 Effect on Receipt

Once all *ConfigureNotificationFlags* commands have been received, a fully populated scheme will be available, and two way mirroring can then be enabled.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

7102 D.3.2.3.1.12 GetNotifiedMessage Command

7103 The *GetNotifiedMessage* command is used only when a BOMD is being mirrored. This
7104 command provides a method for the BOMD to notify the Mirror message queue that it wants to
7105 receive commands that the Mirror has queued. The Notification flags set within the command
7106 shall inform the mirror of the commands that the BOMD is requesting.

7107 ***D.3.2.3.1.12.1 Payload Format***

Octets	1	2	4
Data Type	Unsigned 8-bit Integer	Unsigned 16-bit Integer	32-bit BitMap
Field Name	Notification Scheme (M)	Notification Flag attribute ID (M)	Notification Flags #N (M)

7108 **Figure D-35– *GetNotifiedMessage* Command Payload**

7109 ***D.3.2.3.1.12.2 Payload Details***

7110 **Notification Scheme (mandatory):** An unsigned 8-bit integer that allows for the pre-loading of
7111 the Notification Flags bit mapping to ZCL or Smart Energy Standard commands. Figure D-31
7112 details the schemes that are currently supported within the Smart Energy Standard.

7113 **Notification Flag Attribute ID (mandatory):** An unsigned 16-bit integer that denotes the
7114 attribute id of the notification flag (1-8) that is included in this command.

7115 **Notification Flags #N (mandatory):** The Notification Flags attribute/parameter indicating the
7116 command being requested. See D.3.3.2.1.1 and D.3.3.2.1.2 for further details.

7117 ***D.3.2.3.1.12.3 When Generated***

7118 The *GetNotifiedMessage* command is generated in response to the flags that have been set within
7119 the *NotificationFlags#N* attribute/parameter within the *MirrorReportAttributeResponse*
7120 command. The BOMD shall be in control of when it sends this command and what commands it
7121 shall request. This command should only be generated when there is no specific "GET"
7122 command to be used to fetch the information. i.e. if the scheme supports *GetProfile* &
7123 *GetProfileResponse*, the attribute could be configured to inform the BOMD that the mirror
7124 requires some load profile information. Therefore, by setting the flag in this command, the
7125 BOMD is requesting that the *GetProfile* command is now sent to it.

7126 The BOMD may choose not to initiate the process if the battery level does not allow it, or if the
7127 request is sent too often.

7128 ***D.3.2.3.1.12.4 Effect on Receipt***

7129 Dependent on the flags set within the command, the Mirror shall send down the appropriate
7130 command to the BOMD.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

7131

7132 D.3.2.3.1.13 Supply Status Response Command

7133 This command is transmitted by a Metering Device in response to a *Change Supply* command.

7134 ***D.3.2.3.1.13.1 Payload Format***

Octets	**4**	**4**	**4**	**1**
Data Type	Unsigned 32-bit Integer	Unsigned 32 – bit Integer	UTC Time	8-bit Enumeration
Field Name	Provider ID (M)	Issuer Event ID (M)	Implementation Date/Time (M)	Supply Status (after implementation) (M)

7135 **Figure D-36– Supply Status Response Command Payload**

7136 ***D.3.2.3.1.13.2 Payload Details***

7137 **Provider ID (mandatory):** An unsigned 32-bit field containing a unique identifier for the
7138 commodity provider to whom this command relates.

7139 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. When
7140 new information is provided that replaces older information for the same time period, this field
7141 allows devices to determine which information is newer. The value contained in this field is a
7142 unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)
7143 identifying when the command was issued. Thus, newer information will have a value in the
7144 *Issuer Event ID* field that is larger than older information.

7145 **Implementation Date/Time (mandatory):** A UTC Time field to indicate the date at which the
7146 originating command was to be applied.

7147 **Supply Status (mandatory):** An 8-bit enumeration field indicating the status of the energy
7148 supply controlled by the Metering Device following implementation of the originating command.
7149 The enumerated values for this field are outlined in Table D-56.

7150 **Table D-56– Supply Status Field Enumerations**

Enumerated Value	**Status**
0x00	Supply OFF
0x01	Supply OFF / ARMED
0x02	Supply ON

7151

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

7152 ***D.3.2.3.1.13.3 When Generated***

7153 This command is transmitted by a Metering Device to indicate that a *Change Supply* command
7154 has been successfully executed. It shall be sent if an acknowledgment is requested in the
7155 originating command (see sub-clause D.3.3.3.1.12).

7156

7157 D.3.2.3.1.14 Start Sampling Response Command

7158 This command is transmitted by a Metering Device in response to a *StartSampling* command.

7159 ***D.3.2.3.1.14.1 Payload Format***

7160 The *StartSamplingResponse* command payload shall be formatted as illustrated in Figure D-39

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

7161 **Figure D-37– *StartSamplingResponse* Command Payload**

7162 ***D.3.2.3.1.14.2 Payload Details***

7163 **Sample ID:** 16 Bit Unsigned Integer indicating the ID allocated by the Metering Device for the
7164 requested Sampling session. If the Metering Device is unable to support a further Sampling
7165 session, *Sample ID* shall be returned as 0xFFFF. If valid, the *Sample ID* shall be used for all
7166 further communication regarding this Sampling session.

7167 NOTE that the Metering Device may reserve a *Sample ID of 0x0000* in order to provide an
7168 alternative mechanism for retrieving Profile data. This mechanism will allow an increased
7169 number of samples to be returned than is available via the existing (automatically started) Profile
7170 mechanism.

7171

7172 D.3.3 Client

7173 D.3.3.1 Dependencies

7174 No additional dependencies.

7175 D.3.3.2 Attributes

7176 For convenience, the attributes defined in this specification are arranged into sets of related
7177 attributes; each set can contain up to 256 attributes. Attribute identifiers are encoded such that the

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

most significant Octet specifies the attribute set and the least significant Octet specifies the attribute within the set. The currently defined attribute sets are listed in Table D-57.

Table D-57 – Metering Cluster Client Attribute Sets

Attribute Set Identifier	Description
0x00	Notification Attribute Set
0x01 to 0xFF	Reserved

D.3.3.2.1 **Notification Attribute Set**

The Notification Attribute Set is used to notify battery operated mirrored devices (BOMDs) that the ESI or other HAN device has pending information which should be fetched.

Only clients on a mirror endpoint shall support this attribute set.

When commands / attributes are received into the ESI from the HES or other HAN devices, the ESI will store the corresponding information and set the appropriate bits in the Notification Flag attributes (BitMaps). The ESI shall reset the bit as soon as a '*Get'* command with the corresponding message type is received and all commands of the appropriate type have been retrieved (this is to allow for multiple commands of the same type).

Table D-58 – Notification Attribute Set

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0000	FunctionalNotificationFlags	32 bit BitMap	0x00000000 - 0xFFFFFFFF	Read	0	O
0x0001	NotificationFlags2	32 bit BitMap	0x00000000 - 0xFFFFFFFF	Read	0	O
0x0002	NotificationFlags3	32 bit BitMap	0x00000000 - 0xFFFFFFFF	Read	0	O
0x0003	NotificationFlags4	32 bit BitMap	0x00000000 - 0xFFFFFFFF	Read	0	O
0x0004	NotificationFlags5	32 Bit BitMap	0x00000000 - 0xFFFFFFFF	Read	0	O
0x0005	NotificationFlags6	32 bit BitMap	0x00000000 - 0xFFFFFFFF	Read	0	O
0x0006	NotificationFlags7	32 bit BitMap	0x00000000 - 0xFFFFFFFF	Read	0	O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0007	NotificationFlags8	32 bit BitMap	0x00000000 - 0xFFFFFFFF	Read	0	O
0x0008 - 0x00FF	Reserved					

7192

7193 D.3.3.2.1.1 FunctionalNotificationFlags

7194 The *FunctionalNotificationFlags* attribute is implemented as a set of bit flags which are have a
7195 predefined action associated with a bit that is not based on a specific command, but may require
7196 the Mirrored device to trigger some additional functionality within the system. The Bit Flags are
7197 defined as shown below:

7198 **Table D-59 – Functional Notification Flags**

Bit Number	Waiting Command
0	New OTA Firmware
1	CBKE Update Request
2	Time Sync
3	Reserved
4	Stay Awake Request HAN
5	Stay Awake Request WAN
6-8	Push Historical Metering Data Attribute Set
9-11	Push Historical Prepayment Data Attribute Set
12	Push All Static Data - Basic Cluster
13	Push All Static Data - Metering Cluster
14	Push All Static Data - Prepayment Cluster
15	NetworkKeyActive
16	Display Message
17	Cancel All Messages
18	Change Supply
19	Local Change Supply
20	SetUncontrolledFlowThreshold
21	Tunnel Message Pending
22	Get Snapshot
23	Get Sampled Data
24-31	Reserved

7199

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

New OTA Firmware Flag: will be set by the ESI, when the ESI has new OTA Firmware to send to the BOMD. The BOMD can then make the decision on when it starts the OTA upgrade request.

CBKE Update Request Flag: requests the BOMD to initiate the CBKE process with the Trust Center to replace the link key currently in use.

Time Sync Request Flag: requests the BOMD to initiate the time synchronization process with the Time server.

The Stay Awake Request Flags: will be set by the ESI when the ESI wants to send a command. There are two types of the Stay Awake requests, one for HAN requests and one for WAN requests; an implementation may react differently depending on the source of the request:

HAN requests: The HAN Stay Awake flag should only be used for commands that originate from HAN and not from the ESI that is supporting the commodity of the Mirrored device.

WAN requests: The WAN Stay Awake flag should only be used for commands that originate from the backhaul network.

Table D-60 – Example Usage of Stay Awake Request Flags

Waiting Command
Schedule Snapshot
Take Snapshot
Start Logging
Get Logging
Get Profile
GetEventLog
ClearEventLog
Reset Demand limit Counter
Read Attribute
Write Attribute

Table D-60 shows example usage of the Stay Awake Request flags. The most likely use is when profile data or snapshots are required. The commands shown require additional parameters to be sent by the requesting device to solicit the correct response from the receiver.

A battery operated meter should read the Notification Flags regularly or enable the *MirrorReportAttributeResponse* command. If the *StayAwakeRequest* flag is set, the battery operated meter shall poll its parent node at least three times for pending messages. The polling interval shall be configurable and not less than 250 ms.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

The ESI shall try to send commands after the BOMD has pushed meter readings or read the Notification Flags. After reception of a command, the BOMD shall read the *NotificationFlags* again. If the ESI has successfully transmitted all pending commands, it shall reset the StayAwakeRequest flag.

Nevertheless, the BOMD can decide to go to sleep if the *StayAwakeRequest* flag is not reset after consecutive reads of the *NotificationFlags* attribute or if it is required by its power supply constraints.

Push Historical Metering Data Attribute Set: This notification flag requests the BOMD to push a sub set of the historical consumption information found within the Metering cluster's 'Historical Consumption Attribute Set'. The format of the bits is defined within Table D-61.

Table D-61 – Push Historical Metering Data Definition

Bit8	Bit7	Bit6	Description
0	0	1	The Meter shall push up the attributes that relate to the "Day" from the Metering cluster and that the device supports
0	1	0	The Meter shall push up the attributes that relate to the "Week" from the Metering cluster and that the device supports
1	1	0	The Meter shall push up the attributes that relate to the "Month" from the Metering cluster and that the device supports
1	1	1	The Meter shall push up the attributes that relate to the "Year" from the Metering cluster and that the device supports

Push Historical Payment Data attribute Set: This notification flag requests the BOMD to push a sub set of the historical consumption cost information found within the Prepayment cluster's 'Historical Cost Consumption Attribute Set'. The format of the bits is defined within Table D-62.

Table D-62 – Push Historical Payment Data Attribute Definition

Bit11	Bit10	Bit9	Description
0	0	1	The Meter shall push up the attributes that relate to the "Day" from the Prepayment cluster and that the device supports
0	1	0	The Meter shall push up the attributes that relate to the "Week" from the Prepayment cluster and that the device supports
1	1	0	The Meter shall push up the attributes that relate to the "Month" from the Prepayment cluster and that the device supports
1	1	1	The Meter shall push up the attributes that relate to the "Year" from the Prepayment cluster and that the device supports

Push All Static Data - Basic Cluster: This notification flag requests the BOMD to push all of the attributes within the Basic cluster that are supported by the mirrored meter.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Push all static Data - Metering Cluster: This notification flag requests the BOMD to push all of the attributes within the Metering cluster that are supported by the mirrored meter.

Push All Static Data - Prepayment Cluster: This notification flag requests the BOMD to push all of the attributes within the Prepayment cluster that are supported by the mirrored meter.

Network Key Active: When this notification flag has been set, the meter shall check with the TC to update the network key.

Display Message: When this notification flag has been set, the meter shall send a *Get Last Message* command to the associated Messaging cluster server (see D.5.3.3.1 for further details).

Cancel All Messages: When this notification flag has been set, the meter shall send a *GetMessageCancellation* command to the associated Messaging cluster server (see D.5.3.3.3 for further details).

Change Supply Message: When this notification flag has been set, the meter shall send a *GetNotifiedMessage* command to the mirror.

Local Change Supply Message: When this notification flag has been set, the meter shall send a *GetNotifiedMessage* command to the mirror.

SetUncontrolledFlowThreshold Message: When this notification flag has been set, the meter shall send a *GetNotifiedMessage* command to the mirror.

Tunnel Message Pending: When set, this notification flag indicates to the BOMD that a message is pending retrieval via the tunnel. If any message(s) is/are pending, then the flag shall be cleared when the last pending message is retrieved.

Get Snapshot Message: When this notification flag has been set, the meter shall send a *GetNotifiedMessage* command to the mirror.

Get Sampled Data Message: When this notification flag has been set, the meter shall send a *GetNotifiedMessage* command to the mirror.

D.3.3.2.1.2 NotificationFlags Attributes

NotificationFlags2 to NotificationFlags8 are 32-bit bitmaps that each represent a series of flags. Each flag represents an outstanding command that the Mirror is holding on behalf of the BOMD. Each flag represents a different command. The format of these attributes is dictated by the scheme that is currently in operation.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

7275

7276 ### D.3.3.3 Client Commands

7277 D.3.3.3.1 **Commands Generated**

7278 The command IDs generated by the Metering client cluster are listed in Table D-63.

7279

Table D-63– Generated Command IDs for the Metering Client

Command Identifier Field Value	Description	Mandatory / Optional
0x00	*Get Profile*	O
0x01	*Request Mirror Response*	O
0x02	*Mirror Removed*	O
0x03	*Request Fast Poll Mode*	O
0x04	*ScheduleSnapshot*	O
0x05	*TakeSnapshot*	O
0x06	*GetSnapshot*	O
0x07	*StartSampling*	O
0x08	*GetSampledData*	O
0x09	*MirrorReport AttributeResponse*	O
0x0A	*ResetLoadLimit Counter*	O
0x0B	*Change Supply*	O
0x0C	*Local Change Supply*	O
0x0D	*SetSupplyStatus*	O
0x0E	*SetUncontrolledFlowThreshold*	O
0x0F-0xFF	Reserved	

7280

7281 D.3.3.3.1.1 Get Profile Command

7282 The *Get Profile* command payload shall be formatted as illustrated in Figure D-38.

7283

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Octets	1	4	1
Data Type	8-bit Enumeration[a]	UTCTime	Unsigned 8-bit Integer
Field Name	Interval Channel	End Time	NumberOfPeriods

a. CCB 1077

Figure D-38– *Get Profile* Command Payload

D.3.3.3.1.1.1 Payload Details

Interval Channel: Enumerated value used to select the quantity of interest returned by the *GetProfileReponse* command. The Interval Channel values are listed in Table D-64.

Table D-64– Interval Channel Values

Enumerated Value	Description
0	Consumption Delivered
1	Consumption Received
2 to 255	Not used

EndTime: 32-bit value (in UTCTime) used to select an Intervals block from all the Intervals blocks available. The Intervals block returned is the most recent block with its EndTime equal or older to the one provided. The most recent Intervals block is requested using an End Time set to 0x00000000, subsequent Intervals block are requested using an End time set to the EndTime of the previous block - (number of intervals of the previous block * ProfileIntervalPeriod).

NumberofPeriods: Represents the number of intervals being requested. This value cannot exceed the size stipulated in the *MaxNumberOfPeriodsDelivered* attribute. If more intervals are requested than can be delivered, the GetProfileResponse will return the number of intervals equal to *MaxNumberOfPeriodsDelivered*. If fewer intervals are available for the time period, only those available are returned.

D.3.3.3.1.1.2 When Generated

The *GetProfile* command is generated when a client device wishes to retrieve a list of captured Energy, Gas or water consumption for profiling purposes. Due to the potentially large amount of profile data available, the client device should store previously gathered data and only request the most current data. When initially gathering significant amounts of historical interval data, the *GetProfile* command should not be issued any more frequently than 7.5 seconds to prevent overwhelming the ZigBee network.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

7309 ***D.3.3.3.1.1.3 Command Processing Response***

7310 If failure occurs in recognizing or processing the payload of the *GetProfile* command, the
7311 appropriate enumerated ZCL status (as referenced in the ZCL Cluster Library specification)
7312 will be returned. On success, a non-Default Response is returned without a ZCL status code.

7313 ***D.3.3.3.1.1.4 Effect on Receipt***

7314 On receipt of this command, the device shall send a *GetProfileReponse* command (see sub-clause
7315 D.3.2.3.1.1).

7316

7317 D.3.3.3.1.2 Request Mirror Response Command

7318 The *Request Mirror Response* Command allows the ESI to inform a sleepy Metering Device it
7319 has the ability to store and mirror its data.

7320 ***D.3.3.3.1.2.1 Payload Format***

7321 The *Request Mirror Response* command payload shall be formatted as illustrated in Figure D-39

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

7322 **Figure D-39– *Request Mirror Response* Command Payload**

7323 ***D.3.3.3.1.2.2 Payload Details***

7324 **EndPoint ID:** 16 Bit Unsigned Integer indicating the End Point ID to contain the Metering
7325 Devices meter data. Valid End Point ID values are 0x0001 to 0x00F0. If the ESI is able to mirror
7326 the Metering Device data, the low byte of the unsigned 16 bit integer shall be used to contain the
7327 eight bit EndPoint ID. If the ESI is unable to mirror the Metering Device data, EndPoint ID
7328 shall be returned as 0xFFFF. All other EndPoint ID values are reserved. If valid, the Metering
7329 device shall use the EndPoint ID to forward its metered data.

7330

7331 D.3.3.3.1.3 Mirror Removed Command

7332 The *Mirror Removed* Command allows the ESI to inform a sleepy Metering Device mirroring
7333 support has been removed or halted.

7334 ***D.3.3.3.1.3.1 Payload Format***

7335 The *Mirror Removed* command payload shall be formatted as illustrated in Figure D-40:

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

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

Figure D-40– *Mirror Removed* Command Payload

D.3.3.3.1.3.2 Payload Details

Removed EndPoint ID: 16 Bit Unsigned Integer indicating the End Point ID previously containing the Metering Devices meter data.

D.3.3.3.1.4 Request Fast Poll Mode Command

D.3.3.3.1.4.1 Payload Format

The *Request Fast Poll Mode* shall be formatted as illustrated in Figure D-41:

Octets	1	1
Data Type	Unsigned 8-bit Integer	Unsigned 8-bit Integer
Field Name	Fast Poll Update Period (seconds)	Duration (minutes)

Figure D-41– *Request Fast Poll Mode* Command Payload

D.3.3.3.1.4.2 Payload Details

Fast Poll Update Period: Desired fast poll period not to be less than the *FastPollUpdatePeriod* attribute.

Duration: Desired duration for the server to remain in fast poll mode not to exceed 15 minutes as specified in sub-clause D.3.4.2.

D.3.3.3.1.4.3 When Generated

The *Request Fast Poll Mode* command is generated when the metering client wishes to receive near real-time updates of *InstantaneousDemand*. Fast poll mode shall only be requested as a result of user interaction (for example, the pushing of a button or activation of fast poll mode by a menu choice).

D.3.3.3.1.4.4 Effect on Receipt

The metering device may continuously update *InstantaneousDemand* as measurements are acquired, but at a minimum *InstantaneousDemand* must be updated at the end of each *FastPollUpdatePeriod*.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

7359 D.3.3.3.1.5 ScheduleSnapshot Command

7360 This command is used to set up a schedule of when the device shall create snapshot data. See
7361 section D.3.4.5 for further details. It is recommended that schedules are persisted across a reboot.

7362 ***D.3.3.3.1.5.1 Payload Format***

Octets	**4**	**1**	**1**	**Variable**
Data Type	Unsigned 32 –bit Integer	Unsigned 8-bit Integer	Unsigned 8-bit Integer	
Field Name	Issuer Event ID (M)	Command Index (M)	Total Number of Commands (M)	Snapshot Schedule Payload (M)

7363 **Figure D-42– *ScheduleSnapshot* Command Payload**

7364 ***D.3.3.3.1.5.2 Payload Details***

7365 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. When
7366 new information is provided that replaces older information for the same time period, this field
7367 allows devices to determine which information is newer. The value contained in this field is a
7368 unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)
7369 identifying when the snapshot command was issued. Thus, newer information will have a value
7370 in the Issuer Event ID field that is larger than older information.This is required when the
7371 snapshot data needs to be transmitted over several messages, allowing for the client to easily
7372 identify the set of messages that form a group.

7373 **Command Index (mandatory):** The *CommandIndex* is used to count the payload fragments for
7374 the case where the entire payload does not fit into one message. The *CommandIndex* starts at 0
7375 and is incremented for each fragment belonging to the same command.

7376 **Total Number of Commands (mandatory):** In the case where the entire payload does not fit
7377 into one message, the *Total Number of Commands* field indicates the total number of sub-
7378 commands in the message.

7379 **SnapshotSchedulePayload (mandatory):**

1	**4**	**3**	**1**	**4**
Unsigned 8-bit Integer	UTCTime	Unsigned 24-bit BitMap	Unsigned 8-bit Enumeration	32-bit BitMap
Snapshot Schedule ID (M)	Snapshot Start Time (M)	Snapshot Schedule (M)	Snapshot Payload Type (M)	Snapshot Cause (M)

7380 **Figure D-43–*SnapshotSchedulePayload* Format**

7381 **Snapshot Schedule ID (mandatory):** The unique ID of the Snapshot schedule; a range
7382 of 1-254 is supported, denoting a maximum of 254 different schedules that could be set
7383 up within the device.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Snapshot Start Time (mandatory): The *Snapshot Start Time* denotes the date/time when the Snapshot schedule is to start.

Snapshot Schedule (mandatory): A 24-bit value indicating the schedule that should be used for the snapshot. The snapshot schedule bit field is formatted as indicated in Table D-65.

Table D-65– Snapshot Schedule BitMap

Bit	Description
0-19	The frequency that the snapshot should be taken in. The format of the duration is defined by bits 20-21
20-21	Frequency Type of the Snapshot 00 = Day 01 = Week 10 = Month 11 = Reserved
22-23	Wild-card Frequency of the Snapshot 00 = Start of 01 = End of 10 = Wild-card not used 11 = Reserved

SnapshotPayloadType (mandatory): The *SnapshotPayloadType* is an 8-bit enumerator defining the format of the *SnapshotPayload* required. The different snapshot types are listed in Table D-53. The server selects the *SnapshotPayloadType* based on the charging scheme in use.

Snapshot Cause (mandatory): A 32-bit BitMap indicating the cause of the snapshot. The snapshot cause values are listed in Table D-52.

D.3.3.3.1.6 TakeSnapshot Command

This command is used to instruct the cluster server to take a single snapshot. See section D.3.4.5 for further details.

D.3.3.3.1.6.1 Payload Format

Octets	4
Data Type	32-bit BitMap
Field Name	Snapshot Cause (M)

Figure D-44– *TakeSnapshot* Command Payload

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

7402 ***D.3.3.3.1.6.2 Payload Details***

7403 **Snapshot Cause (mandatory):** A 32-bit BitMap indicating the cause of the snapshot. The
7404 snapshot cause values are listed in Table D-52. Note that the *Manually Triggered from Client*
7405 flag shall additionally be set for all Snapshots triggered in this manner.

7406 ***D.3.3.3.1.6.3 Effect on Receipt***

7407 On receipt of this command, the server shall take and store a snapshot with cause 10 (Manually
7408 Triggered from Client) set in addition to the requested cause (see Table D-52).

7409

7410 D.3.3.3.1.7 GetSnapshot Command

7411 This command is used to request snapshot data from the cluster server. See section D.3.4.5 for
7412 further details.

7413 ***D.3.3.3.1.7.1 Payload Format***

Octets	**4**	**4**	**1**	**4**
Data Type	UTC Time	UTC Time	Unsigned 8-bit Integer	32-bit BitMap
Field Name	Earliest Start Time (M)	Latest End Time (M)	Snapshot Offset (M)	Snapshot Cause (M)

7414 **Figure D-45– *GetSnapshot* Command Payload**

7415 ***D.3.3.3.1.7.2 Payload Details***

7416 **Earliest Start Time (mandatory):** A UTC Timestamp indicating the earliest time of a snapshot
7417 to be returned by a corresponding *Publish Snapshot* command. Snapshots with a time stamp
7418 equal to or greater than the specified *Earliest Start Time* shall be returned.

7419 **Latest End Time (mandatory):** A UTC Timestamp indicating the latest time of a snapshot to be
7420 returned by a corresponding *Publish Snapshot* command. Snapshots with a time stamp less than
7421 the specified *Latest End Time* shall be returned.

7422 **Snapshot Offset (mandatory):** Where multiple snapshots satisfy the selection criteria specified
7423 by the other fields in this command, this field identifies the individual snapshot to be returned.
7424 An offset of zero (0x00) indicates that the first snapshot satisfying the selection criteria should be
7425 returned, 0x01 the second, and so on.

7426 **Snapshot Cause (mandatory):** This field is used to select only snapshots that were taken due to
7427 a specific cause. The allowed values are listed in Table D-52. Setting this field to 0xFFFFFFFF
7428 indicates that all snapshots should be selected, irrespective of the cause.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

7429 ***D.3.3.3.1.7.3 Effect on Receipt***

7430 On receipt of this command, the server shall respond with one or more *Publish Snapshot*
7431 commands representing the first (or next) snapshot meeting the selection criteria and *Snapshot*
7432 *Offset* value detailed in this command. Details of the *Publish Snapshot* command are detailed in
7433 sub-clause D.3.2.3.1.7.

7434 A ZCL Default Response with status NOT_FOUND shall be returned if the server does not have
7435 a snapshot which satisfies the received parameters (e.g. no snapshot with a timestamp between
7436 the *Earliest Start Time* and the *Latest End Time*).

7437

7438 D.3.3.3.1.8 StartSampling Command

7439 The sampling mechanism allows a set of samples of the specified type of data to be taken,
7440 commencing at the stipulated start time. This mechanism may run concurrently with the
7441 capturing of profile data, and may refer to the same parameters, albeit possibly at a different
7442 sampling rate.

7443 ***D.3.3.3.1.8.1 Payload Format***

Octets	**4**	**4**	**1**	**2**	**2**
Data Type	Unsigned 32–bit Integer	UTC Time	8-bit Enumeration	Unsigned 16-bit Integer	Unsigned 16-bit Integer
Field Name	Issuer Event ID (M)	StartSampling Time (M)	SampleType (M)	SampleRequest Interval (M)	MaxNumberof Samples (M)

7444 **Figure D-46– *StartSampling* Command Payload**

7445 ***D.3.3.3.1.8.2 Payload Details***

7446 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. When
7447 new information is provided that replaces older information for the same time period, this field
7448 allows devices to determine which information is newer. The value contained in this field is a
7449 unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)
7450 identifying when the command was issued. Thus, newer information will have a value in the
7451 Issuer Event ID field that is larger than older information. Commands should be ignored if the
7452 value of the *Issuer Event ID* is equal to or less than the previous value; a device MAY return a
7453 ZCL Default Response command in this case[15].

7454 **StartSamplingTime (mandatory):** A UTC Time field to denote the time at which the sampling
7455 should start. A start Date/Time of 0x00000000 shall indicate that the command should be
7456 executed immediately. A start Date/Time of 0xFFFFFFFF shall cause an existing *StartSampling*
7457 command with the same *Issuer Event ID* to be cancelled.

[15] CCB 2010

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

SampleType (mandatory): An 8 bit enumeration that identifies the type of data being sampled. Possible values are defined in Table D-54.

SampleRequestInterval (mandatory): An unsigned 16-bit field representing the interval or time in seconds between samples.

MaxNumberofSamples (mandatory): A 16 bit unsigned integer that represents the number of samples to be taken.

D.3.3.3.1.8.3 Effect on Receipt

On receipt of the *StartSampling* command, the Metering Device shall respond with a *StartSamplingResponse* command indicating the *Sample ID* allocated to this Sampling session. If the Metering Device is unable to support a further Sampling session, *Sample ID* shall be returned as 0xFFFF. See D.3.2.3.1.14 for further details.

D.3.3.3.1.9 GetSampledData Command

This command is used to request sampled data from the server. Note that it is the responsibility of the client to ensure that it does not request more samples than can be held in a single command payload.

D.3.3.3.1.9.1 Payload Format

Octets	**2**	**4**	**1**	**2**
Data Type	Unsigned 16 –bit Integer	UTC Time	8-bit Enumeration	Unsigned 16-bit Integer
Field Name	SampleID (M)	EarliestSampleTime (M)	SampleType (M)	NumberOfSamples (M)

Figure D-47– *GetSampledData* Command Payload

D.3.3.3.1.9.2 Payload Details

SampleID (mandatory): Unique identifier allocated to this Sampling session. This field allows devices to match response data with the appropriate request. See D.3.2.3.1.14 for further details.

EarliestSampleTime (mandatory): A UTC Timestamp indicating the earliest time of a sample to be returned. Samples with a timestamp equal to or greater than the specified *EarliestSampleTime* shall be returned.

SampleType (mandatory): An 8 bit enumeration that identifies the required type of sampled data. Possible values are defined in Table D-54.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

NumberOfSamples (mandatory): Represents the number of samples being requested, This value cannot exceed the size stipulated in the *MaxNumberofSamples* field in the *StartSampling* command. If more samples are requested than can be delivered, the *GetSampledDataResponse* command will return the number of samples equal to the *MaxNumberofSamples* field. If fewer samples are available for the time period, only those available are returned.

D.3.3.3.1.9.3 Effect on Receipt

On receipt of this command, the server shall respond with a *GetSampledDataResponse* command containing the samples meeting the selection criteria detailed in this command. Details of the *GetSampledDataResponse* command are detailed in sub-clause D.3.2.3.1.8.

A ZCL Default Response with status NOT_FOUND shall be returned if the server does not have sample data which satisfies the received parameters.

D.3.3.3.1.10 MirrorReportAttributeResponse Command

This command is sent in response to the *ReportAttribute* command when the *MirrorReporting* attribute is set.

D.3.3.3.1.10.1 Payload Format

Octets	1	Variable
Data Type	Unsigned 8-bit Integer	32-bit BitMap
Field Name	Notification Scheme (M)	Notification Flags #N (M)

Figure D-48– MirrorReportAttributeResponse Command Payload

D.3.3.3.1.10.2 Payload Details

The payload of this command is defined within the *ConfigureNotificationScheme* command.

Notification Scheme (mandatory): An unsigned 8-bit integer that allows for the pre-loading of the Notification Flags bit mapping to ZCL or Smart Energy Standard commands. Figure D-31 details the schemes that are currently supported within the Smart Energy Standard.

Notification Flags #N (mandatory): see sections D.3.3.2.1.1 and D.3.3.2.1.2.

D.3.3.3.1.10.3 When Generated

The *MirrorReportAttributeResponse* command is generated in response to the *ReportAttribute* command when the *MirrorReporting* attribute is set. The *MirrorReportAttributeResponse* command is sent from the Mirror to the meter.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

7511 ***D.3.3.3.1.10.4 Effect on Receipt***

7512 On receipt of the *MirrorReportAttributeResponse*, the meter shall check the flags contained
7513 within the payload. It is then up to the meter to request any information that is waiting on the
7514 ESI.

7515

7516 D.3.3.3.1.11 ResetLoadLimitCounter Command

7517 The *ResetLoadLimitCounter* command shall cause the *LoadLimitCounter* attribute to be reset
7518 (see D.3.2.2.7.7 for further details).

7519 ***D.3.3.3.1.11.1 Payload Format***

7520

Octets	**4**	**4**
Data Type	Unsigned 32-bit Integer	Unsigned 32-bit Integer
Field Name	Provider ID (M)	Issuer Event ID (M)

7521 **Figure D-49– *ResetLoadLimitCounter* Command Payload**

7522 ***D.3.3.3.1.11.2 Payload Details***

7523 **Provider ID (mandatory):** An unsigned 32-bit field containing a unique identifier for
7524 the commodity provider.

7525 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. This
7526 field allows devices to determine if a new command has been issued. The value contained in this
7527 field is a unique number managed by upstream servers or a UTC based time stamp (UTCTime
7528 data type) identifying when the command was issued. Thus, a newer command will have a value
7529 in the Issuer Event ID field that is larger than previous versions of the command.

7530

7531 D.3.3.3.1.12 Change Supply Command

7532 This command is sent from the Head-end or ESI to the Metering Device to instruct it to change
7533 the status of the valve or load switch, i.e. the supply.

7534 ***D.3.3.3.1.12.1 Payload Format***

Octets	**4**	**4**	**4**	**4**	**1**	**1**
Data Type	Unsigned 32-bit Integer	Unsigned 32 bit Integer	UTC Time	UTC Time	8 bits Enumeration	8-Bit BitMap

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Octets	4	4	4	4	1	1
Field Name	Provider ID (M)	Issuer Event ID (M)	Request Date/ Time (M)	Implementation Date/Time (M)	Proposed Supply Status (after Implementation)	Supply Control Bits

Figure D-50– Change Supply Command Payload

D.3.3.3.1.12.2 Payload Details

Provider ID (mandatory): An unsigned 32-bit field containing a unique identifier for the commodity provider to whom this command relates.

Issuer Event ID (mandatory): Unique identifier generated by the commodity provider. When new information is provided that replaces older information for the same time period, this field allows devices to determine which information is newer. The value contained in this field is a unique number managed by upstream servers or a UTC based time stamp (UTCTime data type) identifying when the command was issued. Thus, newer information will have a value in the *Issuer Event ID* field that is larger than older information.

Request Date/Time (mandatory): A UTC Time field to indicate the date and time at which the supply change was requested.

Implementation Date/Time (mandatory): A UTC Time field to indicate the date at which the supply change is to be applied. An *Implementation Date/Time* of 0x00000000 shall indicate that the command should be executed immediately. An *Implementation Date/Time* of 0xFFFFFFFF shall cause an existing but pending *Change Supply* command with the same *Provider ID* and *Issuer Event ID* to be cancelled (the status of the supply will not change but the *Proposed Change Supply Implementation Time* attribute shall be reset to zero).

Proposed Supply Status (after Implementation): An 8-bit enumeration field indicating the status of the energy supply controlled by the Metering Device following implementation of this command. The enumerated values for this field are outlined in Table D-56.

Supply Control Bits: An 8-bit BitMap where the least significant nibble defines the Supply Control bits, the encoding of which is outlined in Table D-66:

Table D-66– Supply Control Bits

Bits	Description
0	Acknowledge Required
1	Reserved
2	Reserved
3	Reserved

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Acknowledge Required: Indicates that a *Supply Status Response* command is to be sent in response to this command. Note that the *Supply Status Response* command will only be returned to the originator when the *Change Supply* command has been successfully executed.

D.3.3.3.1.12.3 When Generated

A Head-end or ESI may send an INTERRUPT, ARM or (if allowed) RESTORE command to a metering device.

The execution of an INTERRUPT or ARM command may be delayed, as indicated by the *Implementation Date/Time* field; these commands shall only come from a Head-End via an ESI. A subsequent command with a new *Implementation Date/Time* shall override an existing delayed command. A new command with an *Implementation Date/Time* of 0x00000000 shall be executed immediately, but shall not cancel an existing delayed command; to override an existing delayed command with a command to be executed immediately, a command to cancel the existing command should first be sent followed by the new command to be executed immediately (see notes on *Implementation Date/ Time* field in D.3.3.3.1.12.2 for further details).

The addition of credit or selection of Emergency credit shall not cause a delayed INTERRUPT command to be cancelled (these will be cancelled by the Head-End and a new supply control command sent down).

D.3.3.3.1.12.4 Effect on Receipt

If required, a *Supply Status Response* command shall be returned to the originator when the *Change Supply* command has been successfully executed (see D.3.2.3.1.13 for further details).

A ZCL response, indicating 'Unauthorized' (NOT_AUTHORIZED), shall be immediately returned to an originator requesting a supply change that is not allowed in the current application.

A ZCL response, indicating 'Unavailable' (UNSUP_CLUSTER_COMMAND), shall be immediately returned to an originator requesting a supply change by a metering device that is incapable of carrying out the action (e.g. an INTERRUPT command to a metering device that has no contactor).

A ZCL response, indicating INVALID_VALUE, shall be immediately returned to an originator requesting a supply change containing a non-zero Implementation Date/Time that is less than or equal to the current date/time (i.e. is in the past).

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

7590 D.3.3.3.1.13 Local Change Supply Command

7591 This command is a simplified version of the *Change Supply* command, intended to be sent from
7592 an IHD to a meter as the consequence of a user action on the IHD. Its purpose is to provide a
7593 local disconnection/reconnection button on the IHD in addition to the one on the meter.

7594 ***D.3.3.3.1.13.1 Payload Format***

Octets	**1**
Data Type	8 bits Enumeration
Field Name	Proposed Supply Status

7595 **Figure D-51– Local Change Supply Command Payload**

7596 ***D.3.3.3.1.13.2 Payload Details***

7597 **Proposed Supply Status:** An 8-bit enumeration field indicating the status of the energy supply
7598 controlled by the Metering Device following implementation of this command. The enumerated
7599 values for this field are outlined in Table D-67:

7600 **Table D-67– Local Change Supply: Supply Status Field Enumerations**

Enumerated Value	**Description**
0x00	Reserved
0x01	Supply OFF / ARMED
0x02	Supply ON
0x03-0xFF	Reserved

7601 ***D.3.3.3.1.13.3 When Generated***

7602 An IHD may only request an OFF/ARMED or ON status for the supply. This corresponds to a
7603 local disconnection or reconnection (from Armed state) of the supply, similar to what can be
7604 achieved with a button normally present on electricity meters equipped with a contactor.

7605 ***D.3.3.3.1.13.4 Effect on Receipt***

7606 No *Supply Status Response* command shall be returned to the originator.

7607 A ZCL response, indicating 'Unauthorized' (NOT_AUTHORIZED), shall be immediately
7608 returned to an originator requesting a supply change that is not allowed in the current application.

7609 A ZCL response, indicating 'Unavailable' (UNSUP_CLUSTER_COMMAND), shall be
7610 immediately returned to an originator requesting a supply change by a metering device that is
7611 incapable of carrying out the action (e.g. an INTERRUPT command to an electricity meter that
7612 has no contactor or to a gas meter for which this command is not allowed).

7613

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

7614 D.3.3.3.1.14 SetSupplyStatus Command

7615 This command is used to specify the required status of the supply following the occurance of
7616 certain events on the meter. The meter shall check these requirements to understand whether the
7617 supply should be disabled or enabled following one of these events.

7618 ***D.3.3.3.1.14.1 Payload Format***

Octets	**4**	**1**	**1**	**1**	**1**
Data Type	Unsigned 32 – bit Integer	8 bit Enumeration	8 bit Enumeration	8 bit Enumeration	8 bit Enumeration
Field Name	Issuer Event ID (M)	SupplyTamper State (M):	SupplyDepletion State (M):	SupplyUncontrolled FlowState (M):	LoadLimitSupply State (M):

7619 **Figure D-52– SetSupplyStatus Command Payload**

7620 ***D.3.3.3.1.14.2 Payload Details***

7621 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. When
7622 new information is provided that replaces older information for the same time period, this field
7623 allows devices to determine which information is newer. The value contained in this field is a
7624 unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)
7625 identifying when the command was issued. Thus, newer information will have a value in the
7626 *Issuer Event ID* field that is larger than older information.

7627 **SupplyTamperState (mandatory):** The *SupplyTamperState* indicates the required status of the
7628 supply following the detection of a tamper event within the metering device. The enumerated
7629 values for this field are outlined in Table D-68.

7630 **SupplyDepletionState (mandatory):** The *SupplyDepletionState* indicates the required status of
7631 the supply following detection of a depleted battery within the metering device. The enumerated
7632 values for this field are outlined in Table D-68.

7633 **SupplyUncontrolledFlowState (mandatory):** The *SupplyUncontrolledFlowState* indicates the
7634 required status of the supply following detection of an uncontrolled flow event within the
7635 metering device. The enumerated values for this field are outlined in Table D-68.

7636 **LoadLimitSupplyState (mandatory):** The *LoadLimitSupplyState* indicates the required status
7637 of the supply once the device is in a load limit state. The enumerated values for this field are
7638 outlined in Table D-68.

7639 **Table D-68– SetSupplyStatus: Field Enumerations**

Enumerated Value	**Description**
0x00	Supply OFF

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x01	Supply OFF / ARMED
0x02	Supply ON
0x03	Supply UNCHANGED
0x04-0xFF	Reserved

7640

7641 D.3.3.3.1.15 SetUncontrolledFlowThreshold Command

7642 This command is used to update the 'Uncontrolled Flow Rate' configuration data used by flow
7643 meters.

7644 ***D.3.3.3.1.15.1 Payload Format***

Octets	**4**	**4**	**2**	**1**	**2**	**2**
Data Type	Unsigned 32-bit Integer	Unsigned 32 bit Integer	Unsigned 16 bit Integer	8 bits Enumeration	Unsigned 16 bit Integer	Unsigned 16 bit Integer
Field Name	Provider ID (M)	Issuer Event ID (M)	Uncontrolled Flow Threshold (M)	Unit of Measure (M)	Multiplier (M)	Divisor (M)

7645

Octets	**1**	**2**
Data Type	Unsigned 8 bit Integer	Unsigned 16 bit Integer
Field Name	Stabilisation Period (M)	Measurement Period (M)

7646 **Figure D-53– SetUncontrolledFlowThreshold Command Payload**

7647 ***D.3.3.3.1.15.2 Payload Details***

7648 **Provider ID (mandatory):** An unsigned 32-bit field containing a unique identifier for the
7649 commodity provider to whom this command relates.

7650 **Issuer Event ID (mandatory)**: Unique identifier generated by the commodity provider. When
7651 new information is provided that replaces older information for the same time period, this field
7652 allows devices to determine which information is newer. The value contained in this field is a
7653 unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)
7654 identifying when the command was issued. Thus, newer information will have a value in the
7655 *Issuer Event ID* field that is larger than older information.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Uncontrolled Flow Threshold (mandatory): The threshold above which a flow meter (e.g. Gas or Water) shall detect an uncontrolled flow. A value of 0x0000 indicates the feature in unused.

Unit of Measure (mandatory): An enumeration indicating the unit of measure to be used in conjunction with the *Uncontrolled Flow Threshold* attribute. The enumeration used for this field shall match one of the UnitOfMeasure values using a pure binary format as defined in the Metering cluster (see sub-clause D.3.2.2.4.1).

Multiplier (mandatory): An unsigned 16-bit value indicating the multiplier, to be used in conjunction with the *Uncontrolled Flow Threshold* and *Divisor* fields, to determine the true flow threshold value. A value of 0x0000 is not allowed.

Divisor (mandatory): An unsigned 16-bit value indicating the divisor, to be used in conjunction with the *Uncontrolled Flow Threshold* and *Multiplier* fields, to determine the true flow threshold value. A value of 0x0000 is not allowed.

Stabilisation Period (mandatory): An unsigned 8-bit value indicating the time given to allow the flow to stabilize. It is defined in units of tenths of a second.

Measurement Period (mandatory): An unsigned 16-bit value indicating the period over which the flow is measured and compared against the *Uncontrolled Flow Threshold* value. It is defined in units of 1 second.

D.3.4 Metering Application Guidelines

D.3.4.1 Attribute Reporting

Attribute reporting may be used for sending information in the Reading Information, TOU Information, Meter Status, and Historical Consumption attribute sets. Use of the *Report Attribute* command without report configuration may be used for unsolicited notification of an attribute value change. Sleepy devices may have to poll.

D.3.4.2 Fast Polling or Reporting for Monitoring Energy Savings

Client devices, such as an energy gateway, smart thermostat, or in-home displays can monitor changes to energy saving settings within the premises and give users near real time feedback and results. The Metering cluster can support this by using Attribute Reporting and sending updates at a much faster rate for a short period of time. Client devices can also perform a series of Attribute reads to accomplish the same task. In either case, requests or updates shall be limited to a maximum rate of once every two seconds for a maximum period of 15 minutes. These

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

limitations are required to ensure Smart Energy profile based devices do not waste available bandwidth or prevent other operations within the premises.

D.3.4.3 Metering Data Updates

The frequency and timeliness of updating metering data contained in the Metering Cluster attributes and Profile Intervals is up to the individual Metering device manufacturer's capabilities. As a best practice recommendation, updates of the metering data should not cause delivery of the information to end devices more often than once every 30 seconds. End devices should also not request information more often than once every 30 seconds. The Fast Polling attributes and commands shall be used by client devices requesting information more often than once every 30 seconds.

D.3.4.3.1 **Fast Polling Periods**

Since the *DefaultUpdatePeriod* specifies the normal update interval and *FastPollUpdatePeriod* specifies the fastest possible update interval, it is recommended that metering clients read these attributes to determine the optimal normal/fast polling interval and the optimal fast poll period to request. Client devices shall not request data more frequent than *FastPollUpdatePeriod* or the *AppliedUpdatePeriod*.

D.3.4.4 Mirroring

The SE Profile specifies Mirror support in the Metering cluster to store and provide access to data from metering devices on battery power. Devices with resources to support mirroring advertize the capability using the Basic Attribute Physical Environment.

D.3.4.4.1 **Discovery**

The SE standard does not prescribe how Mirroring is implemented. Devices may query the Basic Cluster attribute *PhysicalEnvironment* to determine Mirrored device capacity prior to CBKE (see sub-clause D.3.4.4.2 below). This would allow a battery based end device to discover if an ESI has capacity to mirror data prior to the process of joining the network in a secure manner, thereby reducing retry attempts. This would also enhance the service discovery of the ZDO Match Descriptor that would be used to determine if an endpoint can request the setup and removal of a mirrored Metering cluster. Once a device has joined the network and performed CBKE, it can then request setup of a mirrored metering cluster. ZDO Discovery should be supported to allow HAN devices to discover the mirror endpoints; only active mirror endpoints shall be discoverable. This process may need to be repeated in the case of a Trust Center swap-out (refer to sub-clause 5.4.2.2.3 for further information).

D.3.4.4.2 **Mirror Attributes**

The mandatory *Basic*, *Metering*, and (where applicable) *Prepayment* attributes shall be supported. The Basic Cluster *PhysicalEnvironment* attribute shall be supported on ESIs supporting mirroring functionality; an enumerated value of 0x01 would indicate that the device has the capacity to mirror an end device; a value of 0x00 would specify an "Unspecified

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

environment" per the ZCL specification. Only the Basic cluster for devices capable of providing a mirror shall have the *PhysicalEnvironment* attribute set to 0x01. The *ZCL Report Attribute* command shall be used to push data to the mirror. Only the metering device that has been granted a mirror on a certain endpoint is allowed to push data to that endpoint. The ZCL Not Authorized return status shall be used to provide access control. The use of ZCL Report Configuration shall not be required to generate *Report Attribute* Command.

Manufacturers will design and manufacture devices to meet customer requirement specifications that will state the functionality of the battery powered meter and therefore devices supporting mirroring in the field will also have to support those requirements through an appropriate choice of optional attributes. Battery powered devices will report attributes to the mirror as required by the customer specification. In the event that the mirror is out of memory space or cannot support the attribute it shall respond ATTRIBUTE_UNSUPPORTED back to the battery-powered meter. The same response (ATTRIBUTE_UNSUPPORTED) will be sent to a device querying the mirror for an attribute it doesn't support. A device querying the mirror for an attribute that is supported but not yet available (the battery powered meter hasn't yet sent the attribute) shall receive a response ATTRIBUTE_UNAVAILABLE from the mirror.

D.3.4.4.3 **Two Way Mirror for BOMD**

The primary purpose of a mirror is to present data from a sleepy battery operated mirrored device (BOMD), to a HAN, when communication to the BOMD is not available. However, there is also a need to pass data to the BOMD in these circumstances.

In ZigBee terms, the device providing a mirror has to be a trusted device. There will be APS security between originating device and mirror, and between mirror and BOMD. Messages that require end-to-end security must be secured by other means.

Any device on the HAN wishing to communicate with a BOMD must do so via the mirror. The mirror and the BOMD must support the Notification Attribute Set of the Metering cluster, designed to allow the BOMD to establish if there any messages waiting on the mirror for collection.There are 4 mechanisms provided to allow information destined for a BOMD to be transferred to the BOMD:-

1 For several required actions, the Notification Flag conveys all required information. Many of the bits within the *Functional Notification Flags* attribute utilize this method; the *Push All Static Data - Metering Cluster* bit is an example of this.

2 For those clusters where the BOMD is a client (e.g. Price, Calendar, Device Management), the flags in the Notification Attribute Set allow the BOMD to quickly determine if there is any new information of interest to the BOMD. This information is normally sent from the backhaul (i.e. the Head End System). Upon waking, and having acquired the status of the relevant Notification Flags, the BOMD will fetch those commands matching the set flags by

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

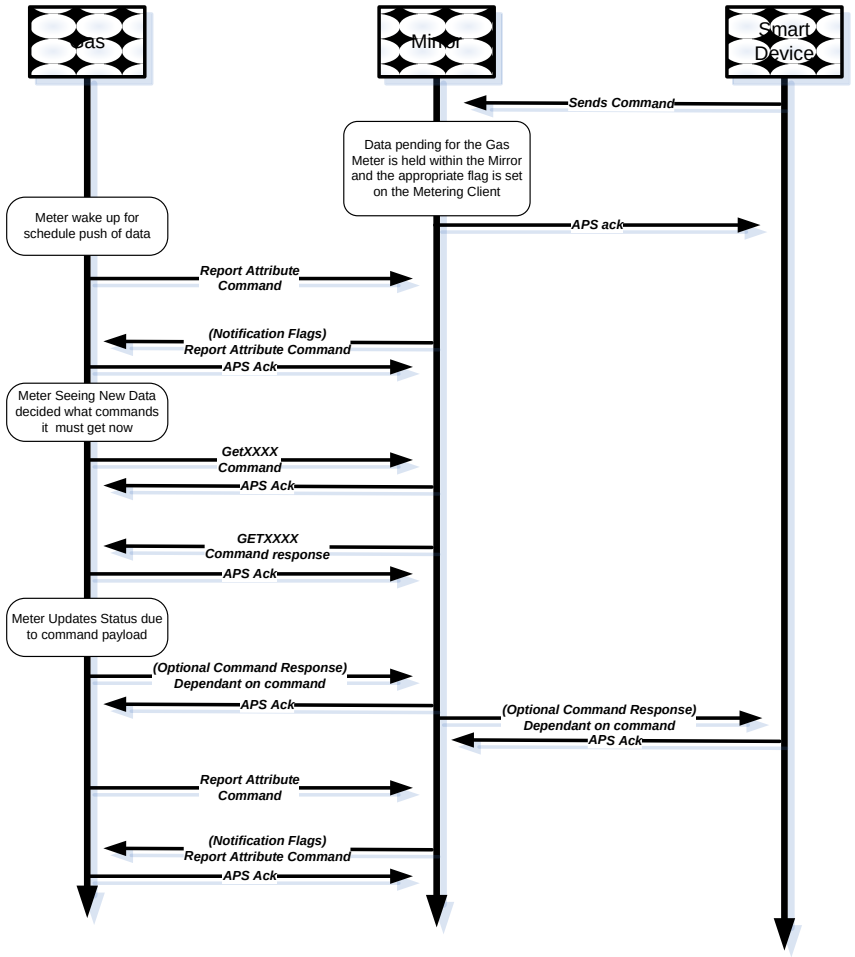
sending the appropriate ‘get’ command(s) **to the associated ESI endpoint**. (Note that the associated data will usually be held on the device providing the ESI; therefore the associated get/publish commands could be utilized multiple times). It is recommended that a ‘binding-type’ mechanism is used internally within the mirroring device to link the ESI and Mirror endpoints.

3 Where a cluster server is located on the BOMD (e.g. Metering, Prepayment and Basic clusters), pre-specified transient commands sent from cluster clients will have to be buffered on the Mirror until such time as the BOMD awakes and can fetch them. Upon waking, and having acquired the status of the relevant Notification Flags, the BOMD will fetch those commands matching the set flags by sending a *GetNotifiedMessage* command **to the Mirror endpoint**. In this case, the Mirror shall remember the address of the device initially originating the command, so that any response can be returned, via the mirror, to that device. As an example, the handling of a (Prepayment cluster) *Consumer Top Up* command utilizes this method.

4 Non-specified transient commands, destined for cluster servers located on a BOMD, will also have to be buffered on the Mirror until such time as the BOMD awakes. In this case, the appropriate ‘Stay Awake’ Notification flag will be set (to advise the BOMD to remain awake for a longer period) and, once the mirroring device recognizes that the BOMD is awake, it shall attempt to push those buffered commands to the BOMD as soon as possible. Attempts to transfer these commands shall be repeated until such time as the command(s) is/are successfully moved to the BOMD (this may not be within the same BOMD wake period). The Mirror shall again remember the address of the device initially originating the command, so that any response can be returned, via the mirror, to that device. As an example, the handling of a (Metering cluster) *Get Profile* command utilizes this method.

.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.



7788

7789 **Figure D-54– Example of Data flow from IHD to Gas meter**

7790 The example in Figure D-54 shows how data is transferred from a HAN device (e.g. IHD) to a
7791 BOMD (e.g. gas meter) via the mirror. There are a number of commands that will be sent from
7792 the IHD to meter, for example:

7793 • Credit Top Up

7794 • Emergency Credit Select

7795 • Local Change Supply

7796 The sequence of events is as follows:

7797 **1** The IHD sends a command to the Mirror

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

2 The mirror "caches" the command and sets the appropriate notification flag, to signal that data is waiting. The mirror also returns a ZCL Default Response to the initiating device with a status code of NOTIFICATION_PENDING. If the command buffer on the mirror is already full, the mirror shall instead return a ZCL Default Response to the initiating device with a status code of INSUFFICIENT_SPACE.

3 The gas meter wakes up and polls for a notification

4 The notification is returned

5 This may be all that is required (e.g. a request to update static data on the mirror).

6 The meter requests the data according to the Notification flag that was raised

7 The mirror sends the command that was originally received from the IHD

8 The meter may update data on the mirror in order to indicate to the device intitiating the command that its action has been carried out.

A mirror that caches a command on behalf of a HAN device, prior to that command being sent to a BOMD, may choose to time out that command after an appropriate period of time. The timeout period may be configurable based on the operator of the network and is not defined by this specification. If the mirror chooses to timeout a cached command, then it shall send a ZCL Default Response to the originator of the message with the same Transaction Sequence Number as received in the cached command. The Default Response shall contain the status code of TIMEOUT.

D.3.4.4.3.1 Responses to an Initiating Device

Commands that have been buffered on a Mirror may trigger ZCL Default or command-specific responses to be returned once the command has reached and been actioned by the BOMD. In turn, these responses should be communicated to the device originally initiating the buffered command.

In order to ensure that these responses are correctly relayed back to the device initiating the original command, it is recommended that:-

1 The mirroring device, if supporting Two-way Mirroring, be able to store information that can be used to track the originator of a command stored in the buffer when the command is retrieved by the BOMD. This information should include, as a minimum, the device address, endpoint and Transaction Sequence Number (TSN) of the original command, for the purposes of relaying a response back to the originator. The TSN used when the buffered command is forwarded to the BOMD should also be included in the stored information for the purpose of matching the information to any associated response. The BOMD shall include the TSN of a forwarded buffered command in any associated response. A TSN in the ZCL header is only 8 bits and this may not provide enough information to produce a unique

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

ID (or unique enough); if this is considered to be insufficient information, the mirroring device could also store the cluster and command id.

2 The mirroring device should relay the ZCL payload of the buffered command to the BOMD using a newly generated ZCL command (i.e. using the address and security associated with the BOMD).

3 The payload of a response from the BOMD to the retrieved buffered command, should be relayed to the originating device in a newly generated ZCL response which utilizes the information stored by the mirroring device as detailed in point 1. This shall include the TSN of the command received from the originating HAN device.

4 All commands retrieved by the BOMD using the notification flag mechanism should support a ZCL default response in cases where an explicit response is not defined (this is the normal mode of operation; however the default response can be disabled).

D.3.4.4.3.2 Unsolicited Commands from a BOMD

Where a command is sent unsolicited from a cluster server on a BOMD, the BOMD should publish that command to the mirror and the mirror should then publish that command to all associated client devices that have bound to the respective server on the mirror.

Client devices wishing to receive unsolicited commands published from a BOMD shall bind to the respective server(s) on the BOMD mirror.

D.3.4.4.3.3 Configuring a Two Way Mirror

1 When utilizing a two-way mirror with a BOMD, certain configuration data must be passed from the BOMD to the mirror once the mirror endpoint has been activated:-Under normal circumstances, a BOMD will utilize one of the predefined Notification Schemes that will be pre-loaded onto the mirror. In this case, only a *ConfigureMirror* command will be required; this command will advise the mirror of the reporting interval to be used, the required mechanism to be used to acquire Notification Flag status, and the predefined Notification Scheme to be used.

2 Predefined Notification schemes cannot be modified. If a BOMD wishes to modify an existing predefined scheme, or utilize a new generic or MSP Notification Scheme, then the associated two-way mirror must be configured with information defining the new scheme before that scheme can be used. New generic schemes should use one of the reserved values 0x03–0x80, MSP schemes should use one of the values 0x081-0x0FE (see Figure D-31).

A BOMD can configure a new Notification Scheme on a mirror once that mirror has been created (endpoint known). The BOMD shall send a *ConfigureNotificationScheme* command

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

to the mirror, together with associated *ConfigureNotificationFlags* command(s), before transmitting a *ConfigureMirror* command that utilizes the new Notification Scheme. On receipt of the *ConfigureNotificationScheme* command, the mirror shall store the NotificationScheme information, and wait for the associated *ConfigureNotificationFlags* commands. Until all of the *ConfigureNotificationFlags* commands have been received, the two-way mirror functionality should be disabled. The Notification Flag Order parameter in the *ConfigureNotificationScheme* command will allow the mirror to determine when all of the *ConfigureNotificationFlags* commands have been received.

D.3.4.4.3.4 Predefined Notification Scheme A

Notification Scheme A is a predefined scheme for the order of the bit strings within each of the NotificationFlag#N attributes. See sections D.3.2.3.1.10 and D.3.2.3.1.11 for configuration of other schemes. Refer to section D.3.4.4.3 for details on the usage of these Notification Flags.

D.3.4.4.3.4.1 MirrorReportAttributeResponse Command Format

The format for Notification Scheme A is ***0x0FFFFFFF*** meaning that the first and only Notification flag to be transmitted within the *MirrorReportAttributeResponse* command will be the *FunctionalNotificationFlags* attribute.

***FunctionalNotificationFlags* Attribute:** Defined in section D.3.3.2.1.1.

D.3.4.4.3.5 Predefined Notification Scheme B

Notification Scheme B is a predefined scheme for the order of the bit strings within each of the NotificationFlag#N attributes. See sections D.3.2.3.1.10 and D.3.2.3.1.11 for configuration of other schemes. Refer to section D.3.4.4.3 for details on the usage of these Notification Flags.

D.3.4.4.3.5.1 MirrorReportAttributeResponse Command Format

The format for Notification Scheme B is ***0x01234FFF*** meaning the first Notification flag to be transmitted within the *MirrorReportAttributeResponse* command will be the *FunctionalNotificationFlags* attribute followed by *NotificationFlags2* to *NotificationFlags5*.

***FunctionalNotificationFlags* Attribute:** Defined in section D.3.3.2.1.1.

NotificationFlags2 Attribute: The *NotificationFlags2* attribute shall be configured to support the Price cluster and is implemented as a set of bit flags which are defined as shown below:

Table D-69– Notification Flags 2

Bit Number	Waiting Command

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Bit Number	Waiting Command
0	PublishPrice[a]
1	PublishBlockPeriod
2	PublishTariffInformation
3	PublishConversionFactor
4	PublishCalorificValue
5	PublishCO2Value
6	PublishBillingPeriod
7	PublishConsolidatedBill
8	PublishPriceMatrix
9	PublishBlockThresholds
10	PublishCurrencyConversion
11	Reserved
12	PublishCreditPaymentInfo
13	PublishCPPEvent
14	PublishTierLabels
15	CancelTariff
16-31	Reserved for future expansion

[a] A *Publish Price* command may result from more than one 'Get' command; for clarity, a *GetCurrentPrice* command should be sent when this flag is set, and a *GetScheduledPrices* command MAY also be sent.

NotificationFlags3 Attribute: The *NotificationFlags3* attribute shall be configured to support the Calendar cluster and is implemented as a set of bit flags which are defined as shown below:

Table D-70– Notification Flags 3

Bit Number	Waiting Command
0	PublishCalendar
1	PublishSpecialDays
2	PublishSeasons
3	PublishWeek
4	PublishDay
5	CancelCalendar
6-31	Reserved for future expansion

NotificationFlags4 Attribute: The *NotificationFlags4* attribute shall be configured to support the Prepayment cluster and is implemented as a set of bit flags which are defined as shown below:

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

7910 **Table D-71– Notification Flags 4**

Bit Number	Waiting Command
0	Select Available Emergency Credit
1	Change Debt
2	Emergency Credit Setup
3	Consumer Top Up
4	Credit Adjustment
5	Change Payment Mode
6	Get Prepay Snapshot
7	Get Top Up Log
8	Set Low Credit Warning Level
9	Get Debt Repayment Log
10	Set Maximum Credit Limit
11	Set Overall Debt Cap
12 – 31	Reserved for future expansion

7911
7912 **NotificationFlags5 Attribute:** The *NotificationFlags5* attribute shall be configured to support
7913 the Device Management cluster and is implemented as a set of bit flags which are defined as
7914 shown below:

7915 **Table D-72– Notification Flags 5**

Bit Number	Waiting Command
0	Publish Change of Tenancy
1	Publish Change of Supplier
2	Request New Password 1 Response
3	Request New Password 2 Response
4	Request New Password 3 Response
5	Request New Password 4 Response
6	UpdateSiteID
7	ResetBatteryCounter
8	UpdateCIN
9 - 31	Reserved for future expansion

7916
7917

7918 **NotificationFlags6 Attribute:** This attribute is not supported, with no bits set.

7919 **NotificationFlags7 Attribute:** This attribute is not supported, with no bits set.

7920 **NotificationFlags8 Attribute:** This attribute is not supported, with no bits set.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

7921

7922 ### D.3.4.5 An Introduction to Snapshots

7923 Where a permanent back-haul connection is not guaranteed, there are occasions when the values
7924 of data items need to be frozen for purposes such as consumer billing. The Snapshot mechanism
7925 is provided to satisfy this requirement.

7926 Snapshots can be triggered in a number of ways:-

7927 • Automatically as a result of certain activities (e.g. end of billing period, change of tariff,
7928 change of supplier)

7929 • At pre-defined points using the *ScheduleSnapshot* command (and confirmed via a
7930 *ScheduleSnapshotResponse* command)

7931 • As a manual/one-off action using the *TakeSnapshot* command (and confirmed via a
7932 *TakeSnapshotResponse* command)

7933 A *Publish Snapshot* command should be generated whenever a new Snapshot is created. Details
7934 of stored Snapshots can be requested using the *GetSnapshot* command; the content(s) of the
7935 required Snapshot(s) will then be returned using one or more *Publish Snapshot* commands.

7936 It is recommended that Snapshot data is persisted across a reboot.

7937

7938 ### D.3.4.6 Supply Control

7939 The Supply Control functionality allows a Head-end System to remotely control the status of the
7940 valve or contactor within a meter. The states of supply status are necessary due to the safety
7941 requirements in certain countries, these are:

7942 • ON

7943 • OFF

7944 • ARMED

7945 The ARMED state is to allow for a remote restoration of the supply that requires action by the
7946 consumer (such as pressing a button on the meter or the IHD). This is to ensure the supply is not
7947 restored remotely whilst in an unsafe situation. The three corresponding commands derived from
7948 IEC 62055 are:

7949 • RESTORE

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

- INTERRUPT
- ARM

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

D.4 Price Cluster

D.4.1 Overview

The Price Cluster provides the mechanism for communicating Gas, Energy, or Water pricing information within the premises. This pricing information is distributed to the ESI from either the utilities or from regional energy providers. The ESI conveys the information (via the Price Cluster mechanisms) to other Smart Energy devices.

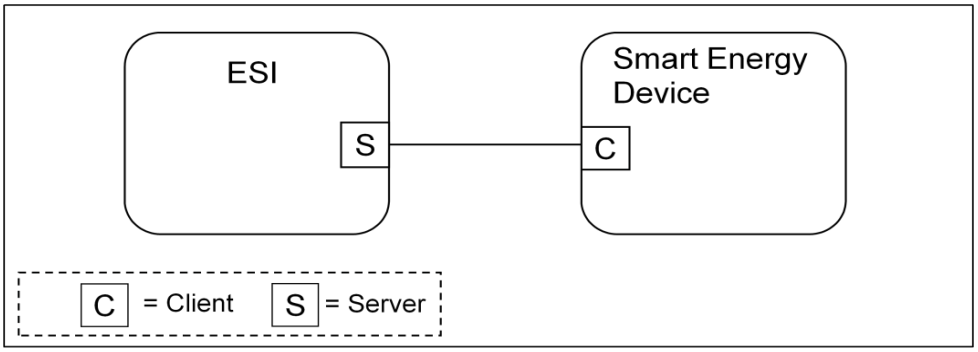


Figure D-55– Price Cluster Client Server Example

Please note the ESI is defined as the Server due to its role in acting as the proxy for upstream price management systems and subsequent data stores.

D.4.2 Server

D.4.2.1 Dependencies

Events carried using this cluster include a timestamp with the assumption that target devices maintain a real time clock. Devices can acquire and synchronize their internal clocks via the ZCL Time server.

If a device does not support a real time clock it is assumed that the device will interpret and utilize the “Start Now” value within the Time field.

D.4.2.2 Attributes

For convenience, the attributes defined in this cluster are arranged into sets of related attributes; each set can contain up to 256 attributes. Attribute identifiers are encoded such that the most significant Octet specifies the attribute set and the least significant Octet specifies the

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

attribute within the set. The currently defined attribute sets are listed in the following Table D-73. The Price Cluster is broken down in to Delivered attribute sets 0x00 to 0x7F and Received attribute sets 0x80 to 0xFF.

Table D-73– Price Cluster Attribute Sets

Attribute Set Identifier	Description
0x00	Tier Label (Delivered)
0x01	Block Threshold (Delivered)
0x02	Block Period (Delivered)
0x03	Commodity
0x04	Block Price Information (Delivered)
0x05	Extended Price Information (Delivered)
0x06	Tariff Information Set (Delivered)
0x07	Billing Information Set (Delivered)
0x08	Credit Payment Attribute Set
0x09 to 0x7F	Reserved
0x80	Received Tier Label
0x81	Received Block Threshold
0x82	Received Block Period
0x83	Reserved
0x84	Received Block Price Information
0x85	Received Extended Price Information
0x86	Received Tariff Information Set
0x87	Received Billing Information Set
0x88 to 0xFF	Reserved

D.4.2.2.1 **Tier Label (Delivered) Set**

Table D-74– Tier Label Attribute Set

Identifier	Name	Type	Length	Access	Default	Mandatory / Optional
0x0000	*Tier1PriceLabel*	Octet String	1 to 13 Octets	Read/ Write	"Tier 1"	O
0x0001	*Tier2PriceLabel*	Octet String	1 to 13 Octets	Read/ Write	"Tier 2"	O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x0002	*Tier3PriceLabel*	Octet String	1 to 13 Octets	Read/ Write	“Tier 3”	O
0x0003	*Tier4PriceLabel*	Octet String	1 to 13 Octets	Read/ Write	“Tier 4”	O
0x0004	*Tier5PriceLabel*	Octet String	1 to 13 Octets	Read/ Write	“Tier 5”	O
0x0005	*Tier6PriceLabel*	Octet String	1 to 13 Octets	Read/ Write	“Tier 6”	O
0x0006	*Tier7PriceLabel*	Octet String	1 to 13 Octets	Read/ Write[a]	“Tier 7”	O
0x0007	*Tier8PriceLabel*	Octet String	1 to 13 Octets	Read/ Write[a]	“Tier 8”	O
0x0008	*Tier9PriceLabel*	Octet String	1 to 13 Octets	Read/ Write[a]	“Tier 9”	O
0x0009	*Tier10PriceLabel*	Octet String	1 to 13 Octets	Read/ Write[a]	“Tier 10”	O
0x000A	*Tier11PriceLabel*	Octet String	1 to 13 Octets	Read/ Write[a]	“Tier 11”	O
0x000B	*Tier12PriceLabel*	Octet String	1 to 13 Octets	Read/ Write[a]	“Tier 12”	O
0x000C	*Tier13PriceLabel*	Octet String	1 to 13 Octets	Read/ Write[a]	“Tier 13”	O
0x000D	*Tier14PriceLabel*	Octet String	1 to 13 Octets	Read/ Write[a]	“Tier 14”	O
0x000E	*Tier15PriceLabel*	Octet String	1 to 13 Octets	Read/ Write[a]	“Tier 15”	O
0x000F	*Tier16PriceLabel*	Octet String	1 to 13 Octets	Read/ Write	“Tier 16”	O
0x0010	*Tier17PriceLabel*	Octet String	1 to 13 Octets	Read/ Write	“Tier 17”	O
0x001n	*TierwxPriceLabel*	Octet String	1 to 13 Octets	Read/ Write	“Tier wx”	O
0x002n	*TieryzPriceLabel*	Octet String	1 to 13 Octets	Read/ Write	“Tier yz”	O
0x2F	*Tier48PriceLabel*	Octet String	1 to 13 Octets	Read/ Write	“Tier 48”	O
0x0030 to 0x00FF	Reserved					

a. CCB 1511

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

7987 D.4.2.2.1.1 TierNPriceLabel Attributes

7988 The *TierNPriceLabel* attributes provide a method for utilities to assign a label to the Price Tier
7989 declared within the *Publish Price* command. The *TierNPriceLabel* attributes are a ZCL Octet
7990 String field capable of storing a 12 character string (the first Octet indicates length) encoded in
7991 the UTF-8 format. Example Tier Price Labels are "Normal", "Shoulder", "Peak", "Real Time"
7992 and "Critical". There are 48 Tier Labels.

7993 Although not prohibited, it is likely (and allowed) that a server will reject an attempt to write to
7994 these attributes; if rejected, the server shall return a ZCL Default Response with a status of either
7995 NOT_AUTHORIZED or READ_ONLY. A client should make provision for a write attempt to
7996 be rejected.[16]
7997

7998 D.4.2.2.2 **Block Threshold (Delivered) Set**

7999 The following set of attributes provides remote access to the Price server Block Thresholds.
8000 Block Threshold values are crossed when the *CurrentBlockPeriodConsumptionDelivered*
8001 attribute value is greater than a *BlockNThreshold* attribute. The number of block thresholds is
8002 indicated by the *Number of Block Thresholds* field in the associated *Publish Pric*e command.
8003 The number of blocks is one greater than the number of thresholds.

8004 **Table D-75– Block Threshold Attribute Set**

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0100	*Block1Threshold*	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0101	*Block2Threshold*	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0102	*Block3Threshold*	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0103	*Block4Threshold*	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0104	*Block5Threshold*	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O

[16] CCB 1511

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x0105	*Block6Threshold*	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0106	*Block7Threshold*	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0107	*Block8Threshold*	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0108	*Block9Threshold*	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0109	*Block10Threshold*	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x010A	*Block11Threshold*	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x010B	*Block12Threshold*	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x010C	*Block13Threshold*	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x010D	*Block14Threshold*	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x010E	*Block15Threshold*	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x010F	*BlockThresholdCount*	Unsigned 8-bit Integer	0x00 to 0xFF	Read Only	-	O
0x0110	*Tier1Block1Threshold*	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x0111	*Tier1Block2Threshold*	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
--	--	--	--	--	--	--
0x011E	*Tier1Block15Threshold*	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x011F	*Tier1BlockThreshold Count*	Unsigned 8-bit Integer	0x00 to 0xFF	Read Only	-	O
0x0120	*Tier2Block1Threshold*	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0121	*Tier2Block2Threshold*	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
--	--	--	--	--	--	--
0x012E	*Tier2Block15Threshold*	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x012F	*Tier2BlockThreshold Count*	Unsigned 8-bit Integer	0x00 to 0xFF	Read Only	-	O
--	--	--	--	--	--	--
0x01FE	*Tier15Block15Threshold*	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x01FF	*Tier15BlockThreshold Count*	Unsigned 8-bit Integer	0x00 to 0xFF	Read Only	-	O

8005

8006 D.4.2.2.2.1 BlockNThreshold

8007 Attributes *Block1Threshold* through *Block15Threshold* represent the block threshold values for a
8008 given period (typically the billing cycle). These values may be updated by the utility on a
8009 seasonal or annual basis. The thresholds are established such that crossing the threshold of
8010 energy consumption for the present block activates the next higher block, which can affect the
8011 energy rate in a positive or negative manner. The values are absolute and always increasing.
8012 The values represent the threshold at the end of a block. The Unit of Measure will be based on the

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

fields defined in the *Publish Price* command, the formatting being defined by attributes within the *Block Period* attribute set.

D.4.2.2.2.2 BlockThresholdCount Attribute

Where a single set of thresholds is used, the *BlockThresholdCount* attribute indicates the number of applicable *BlockNThresholds*. Where more than one set of thresholds is used, each set will be accompanied by an appropriate *TierNBlockThresholdCount* attribute (see D.4.2.2.2.4).

D.4.2.2.2.3 TierNBlockMThreshold Attributes

Attributes *Tier1Block1Threshold* through *Tier15Block15Threshold* represent the block threshold values applicable to a specific TOU tier for a given period (typically the billing cycle). These values may be updated by the utility on a seasonal or annual basis. The thresholds are established such that crossing the threshold of energy consumption for the present block activates the next higher block, which can affect the energy rate in a positive or negative manner. The values are absolute and always increasing. The values represent threshold at the end of a block. The Unit of Measure will based on the fields defined in the *Publish Price* command, the formatting being defined by attributes within the *Block Period* attribute set.

D.4.2.2.2.4 TierNBlockThresholdCount Attributes

The *TierNBlockThresholdCount* attributes hold the number of block thresholds applicable to a given tier. These attributes are used in the case when a combination (TOU/Hybrid) tariff has a separate set of thresholds for each TOU tier. Unused *TierNBlockThresholdCount* attributes shall be set to zero.

D.4.2.2.3 Block Period (Delivered) Set

The following set of attributes provides remote access to the Price server Block Threshold period (typically the billing cycle) information.

Table D-76– Block Period Attribute Set

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory /Optional
0x0200	*StartofBlock Period*	UTCTime	-	Read Only	-	O
0x0201	*BlockPeriod Duration*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0202	*Threshold Multiplier*	Unsigned 24 bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x0203	*Threshold Divisor*	Unsigned 24 bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x0204	*BlockPeriod DurationType*	8-bit BitMap		Read Only	0x00	O
0x0205 to 0x02FF	Reserved					

8037

8038 D.4.2.2.3.1 StartofBlockPeriod Attribute

8039 The *StartofBlockPeriod* attribute represents the start time of the current block tariff period. A
8040 change indicates that a new Block Period is in effect, see sub-clause D.4.4.3 for further details.

8041 D.4.2.2.3.2 BlockPeriodDuration Attribute

8042 The *BlockPeriodDuration* attribute represents the current block tariff period duration in units
8043 defined by the *BlockPeriodDurationType* attribute. A change indicates that only the duration of
8044 the current Block Period has been modified. A client device shall expect a new Block Period
8045 following the expiration of the new duration.

8046 D.4.2.2.3.3 ThresholdMultiplier Attribute

8047 *ThresholdMultiplier* provides a value to be multiplied against Threshold attributes. If present,
8048 this attribute must be applied to all Block Threshold values to derive values that can be
8049 compared against the *CurrentBlockPeriodConsumptionDelivered* attribute within the Metering
8050 cluster (see D.3.2.2.1.13). This attribute must be used in conjunction with the *ThresholdDivisor*
8051 attribute. An attribute value of zero shall result in a unitary multiplier (0x000001).

8052 D.4.2.2.3.4 ThresholdDivisor Attribute

8053 *ThresholdDivisor* provides a value to divide the result of applying the *ThresholdMultiplier*
8054 attribute to Block Threshold values to derive values that can be compared against the
8055 *CurrentBlockPeriodConsumptionDelivered* attribute within the Metering cluster (see
8056 D.3.2.2.1.13). This attribute must be used in conjunction with the *ThresholdMultiplier*
8057 attribute. An attribute value of zero shall result in a unitary divisor (0x000001).

8058 D.4.2.2.3.5 BlockPeriodDurationType Attribute

8059 The *BlockPeriodDurationType* attribute indicates the timebase used for the
8060 *BlockPeriodDuration* attribute. Enumerated values for this attribute are shown in Table D-105. A
8061 default value of 0x00 (Minutes) shall be assumed if this attribute is not present.

8062 D.4.2.2.4 **Commodity Set**

8063 The following set of attributes represents items that are associated with a particular
8064 commodity.

8065

8066 **Table D-77– *Commodity* Attribute Set**

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0300	*CommodityType*	8-bit Enumeration	0x00 to 0xFF	Read Only	-	O
0x0301	*Standing Charge*	Unsigned 32 bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0302	*Conversion Factor*	Unsigned 32 bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	0x10000000	O
0x0303	*Conversion Factor TrailingDigit*	8-bit BitMap		Read Only	0x70	O
0x0304	*Calorific Value*	Unsigned 32 bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	0x2625A00	O
0x0305	*CalorificValue Unit*	8-bit Enumeration		Read Only	0x1	O
0x0306	*Calorific Value TrailingDigit*	8-bit BitMap		Read Only	0x60	O
0x0307 - 0x03FF	Reserved					

D.4.2.2.4.1 CommodityType Attribute

CommodityType provides a label for identifying the type of pricing server present. The attribute is an enumerated value representing the commodity. The defined values are represented by the non-mirrored values (0-127) in the *MeteringDeviceType* attribute enumerations (refer to Table D-27).

D.4.2.2.4.2 Standing Charge Attribute

The value of the *Standing Charge* is a daily fixed charge associated with supplying the commodity, measured in base unit of Currency with the decimal point located as indicated by the Trailing Digits field of a *Publish Price* command (see sub-clause D.4.2.4.1) or *PriceTrailingDigit* attribute (see D.4.2.2.7.7). A value of 0xFFFFFFFF indicates attribute not used.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

8080 D.4.2.2.4.3 ConversionFactor Attribute

8081 The conversion factor is used for gas meter and takes into account changes in the volume of gas
8082 based on temperature and pressure. The *ConversionFactor* attribute represents the current active
8083 value. The *ConversionFactor* is dimensionless. The default value for the *ConversionFactor* is 1,
8084 which means no conversion is applied. A price server can advertise a new/different value at any
8085 time.

8086 D.4.2.2.4.4 ConversionFactorTrailingDigit Attribute

8087 An 8-bit BitMap used to determine where the decimal point is located in the *ConversionFactor*
8088 attribute. The most significant nibble indicates the number of digits to the right of the decimal
8089 point. The least significant nibble is reserved. The *ConversionFactorTrailingDigit* attribute
8090 represents the current active value.

8091 D.4.2.2.4.5 CalorificValue Attribute

8092 The amount of heat generated when a given mass of fuel is completely burned. The
8093 *CalorificValue* is used to convert the measured volume or mass of gas into kWh. The
8094 *CalorificValue* attribute represents the current active value.

8095 D.4.2.2.4.6 CalorificValueUnit Attribute

8096 This attribute defines the unit for the *CalorificValue*. This attribute is an 8-bit enumerated
8097 field. The values and descriptions for this attribute are listed in Table D-78 below. The
8098 *CalorificValueUnit* attribute represents the current active value.

8099 **Table D-78– Values and Descriptions for the *CalorificValueUnit* Attribute**

Values	Description
0x00	Reserved for future use
0x01	MJ/m3
0x02	MJ/kg
0x03 to 0xFF	Reserved for future use

8100

8101 D.4.2.2.4.7 CalorificValueTrailingDigit Attribute

8102 An 8-bit BitMap used to determine where the decimal point is located in the *CalorificValue*
8103 attribute. The most significant nibble indicates the number of digits to the right of the decimal
8104 point. The least significant nibble is reserved. The *CalorificValueTrailingDigit* attribute represents
8105 the current active value.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

8106 D.4.2.2.5 **Block Price Information (Delivered) Set**

8107 The following set of attributes provide remote access to the block prices. The Block Price
8108 Information attribute set supports Block and combined Tier-Block pricing, the number of
8109 blocks is one greater than the number of block thresholds defined in the Pricing cluster.

8110 **Table D-79– Block Price Information Attribute Set**

Identifier	Name	Type	Range	Access	Default	Mandatory /Optional
0x0400	*NoTierBlock1Price*	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x0401	*NoTierBlock2Price*	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x0402	*NoTierBlock3Price*	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x040N	*NoTierBlockN+1Price* ...	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x040F	*NoTierBlock16Price*	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x0410	*Tier1Block1Price*	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x0411	*Tier1Block2Price*	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x0412	*Tier1Block3Price*	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x041N	*Tier1BlockN+1Price ...*	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x041F	*Tier1Block16Price*	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x0420	*Tier2Block1Price*	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x042N	*Tier2BlockN+1Price ...*	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x042F	*Tier2Block16Price*	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x0430	*Tier3Block1Price*	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x043N	*Tier3BlockN+1Price ...*	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x043F	*Tier3Block16Price*	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x0440	*Tier4Block1Price*	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x044N	*Tier4BlockN+1Price ...*	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x044F	*Tier4Block16Price*	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x0450	*Tier5Block1Price*	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x045N	*Tier5BlockN+1Price ...*	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x045F	*Tier5Block16Price*	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x0460	*Tier6Block1Price*	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x046N	*Tier6BlockN+1Price ...*	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x046F	*Tier6Block16Price*	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x0470	*Tier7Block1Price*	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x047N	*Tier7BlockN+1Price ...*	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x047F	*Tier7Block16Price*	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x0480	*Tier8Block1Price*	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x048N	*Tier8BlockN+1Price ...*	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x048F	*Tier8Block16Price*	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x0490	*Tier9Block1Price*	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x049N	*Tier9BlockN+1Price ...*	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x049F	*Tier9Block16Price*	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x04A0	*Tier10Block1Price*	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x04AN	*Tier10BlockN+1Price ...*	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x04AF	*Tier10Block16Price*	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x04B0	*Tier11Block1Price*	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x04BN	*Tier11BlockN+1Price ...*	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x04BF	*Tier11Block16Price*	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x04C0	*Tier12Block1Price*	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x04CN	*Tier12BlockN+1Price* ...	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x04CF	*Tier12Block16Price*	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x04D0	*Tier13Block1Price*	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x04DN	*Tier13BlockN+1Price* ...	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x04DF	*Tier13Block16Price*	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x04E0	*Tier14Block1Price*	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x04EN	*Tier14BlockN+1Price* ...	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x04EF	*Tier14Block16Price*	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x04F0	*Tier15Block1Price*	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x04FN	*Tier15BlockN+1Price* ...	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x04FF	*Tier15Block16Price*	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O

8111
8112

8113 D.4.2.2.5.1 TierNBlockNPrice Attributes

8114 Attributes *PriceNoTierBlock1* through *PriceTier15Block16* represent the price of Energy, Gas,
8115 or Water delivered to the premises (i.e. delivered to the customer from the utility) at a specific

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

price tier as defined by a TOU schedule, Block Threshold or a real time pricing period. If optionally provided, attributes shall be initialized prior to the issuance of associated *Publish Price* commands (see sub- clause D.4.2.4.1). The expected practical limit for the number of *PriceTierNBlockN* attributes supported is 32. The Unit of Measure, Currency and Trailing Digits that apply to this attribute should be obtained from the appropriate fields in a Publish Price command.

D.4.2.2.6 Extended Price Information (Delivered) Set

In case of TOU charging only, the price server allows support for up to 48 TOU rates. To reduce the number of attributes, *Tier1Block1Price* through *Tier15Block1Price* attributes are reused to represent rates for tiers 1 to 15. Rates for tiers 16 to 48 are provided in the extended price information set.

Table D-80– Extended Price Information Set (TOU charging only)

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0500-0x050E	Reserved					
0x050F	PriceTier16	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
…	PriceTierN	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x052F	PriceTier48	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0530-0x05FD	Reserved					
0x05FE	CPP1 Price	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x05FF	CPP2 Price	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O

D.4.2.2.6.1 PriceTierN Attributes

Attributes *PriceTier16* through *PriceTier48* represent the price of Energy, Gas, or Water delivered to the premises (i.e. delivered to the customer from the utility) at a specific price tier.

D.4.2.2.6.2 CPP1 Price Attribute

Attribute *CPP1 Price* represents the price of Energy, Gas, or Water delivered to the premises (i.e. delivered to the customer from the utility) while Critical Peak Pricing ‘CPP1’ is being applied.

D.4.2.2.6.3 CPP2 Price Attribute

Attribute *CPP2 Price* represents the price of Energy, Gas, or Water delivered to the premises (i.e. delivered to the customer from the utility) while Critical Peak Pricing ‘CPP2’ is being applied.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

8138

8139 D.4.2.2.7 **Tariff Information (Delivered) Attribute Set**

8140 The following set of attributes represents items that are associated with a particular Price Tariff.
8141 Please note that the terms tier and rate are used interchangeably here, but do define the same
8142 thing.

8143 **Table D-81– Tariff Information Attribute Set**

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0600-0x060F	Reserved					
0x0610	TariffLabel	Octet String	1 to 25 Octets	Read Only	0	O
0x0611	NumberofPrice TiersInUse	Unsigned 8-bit Integer	0 to 15	Read Only	0	O
0x0612	NumberofBlock ThresholdsInUse	Unsigned 8-bit Integer	0 to 15	Read Only	0	O
0x0613	TierBlockMode	8-bit Enumeration	0x00 to 0xFF	Read Only	0xFF	O
0x0614	Reserved					
0x0615	Unit of Measure	8-bit Enumeration	0x00 to 0xFF	Read Only	0	O
0x0616	Currency	Unsigned 16-bit Integer	0x0000 to 0xFFFF	Read Only	-	O
0x0617	Price Trailing Digit	8-bit BitMap		Read Only	0x00	O
0x0618	Reserved					
0x0619	TariffResolutionPeriod	8-bit Enumeration		Read Only	0	O
0x061A-0x061F	Reserved					
0x0620	CO_2	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	185	O
0x0621	CO_2Unit	8-bit Enumeration		Read Only	1	O
0x0622	CO_2TrailingDigit	8-bit BitMap		Read Only	0	O
0x0628-0x06FF	Reserved					

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

8144 D.4.2.2.7.1 TariffLabel Attribute

8145 The *TariffLabel* attribute provides a method for utilities to assign a label to an entire set of tariff
8146 information. The *TariffLabel* attribute is a ZCL Octet String capable of storing a 24 character
8147 string (the first Octet indicates length) encoded in the UTF-8 format. This attribute is thought of
8148 be useful when a commodity supplier may have multiple tariffs. The *TariffLabel* attribute
8149 represents the current active value.

8150 D.4.2.2.7.2 NumberofPriceTiersInUse Attribute

8151 An 8-bit integer which indicates the number of price tiers used while this tariff is active. Valid
8152 values are from 0 to 48 reflecting block charging only (no price tiers in use) (0) to 48 price tiers
8153 available (48). The *NumberofPriceTiersinUse* attribute represents the current active value.

8154 D.4.2.2.7.3 NumberofBlockThresholdsInUse Attribute

8155 An 8-bit integer which indicates the total number of block thresholds used in the currently active
8156 tariff.

8157 When utilizing TOU charging only, the attribute shall be set to 0 (no thresholds employed).

8158 Where a single set of thresholds is employed, valid values are from 1 to 15 reflecting 1 to 15
8159 block thresholds available. The number of blocks is one greater than the number of block
8160 thresholds.

8161 Where the *TierBlockMode* is set to 2, this attribute indicates the sum of all thresholds employed
8162 for all tiers within the currently active tariff.

8163 D.4.2.2.7.4 TierBlockMode Attribute

8164 An 8-bit enumeration indicating how the mixed TOU / Block charging is to be applied. The value
8165 stored in this attribute is applicable only in the case where *NumberofPriceTiersInUse* is greater
8166 than one and *NumberofBlockThresholdsInUse* is greater than zero. The following table shows
8167 possible values.

8168 **Table D-82– TierBlockMode Enumeration**

Values	Description
0x00	This tariff employs a single set of thresholds. All commodity consumption within a block period is summed and the result compared against the thresholds to determine the Current Block. Each TOU tier will have prices for each block, the current TOU price being dependant on the value of the Current Block. See Figure D-56.
0x01	This tariff employs a single set of thresholds common across all TOU tiers. During a block period, commodity consumption whilst in each TOU tier is individually summed. The summation for a particular TOU tier is compared against the common thresholds to determine the current block. See Figure D-57.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Values	Description
0x02	This combination tariff employs an individual set of Thresholds for each TOU tier. During a block period, commodity consumption whilst in each TOU tier is individually summed. The summation for a particular TOU tier is compared against the thresholds for that tier to determine the current block. This is similar in operation to that shown in Figure D-57 with the exception that the thresholds used can vary from tier to tier.
0x03 to 0xFE	Reserved for future use
0xFF	Not Used

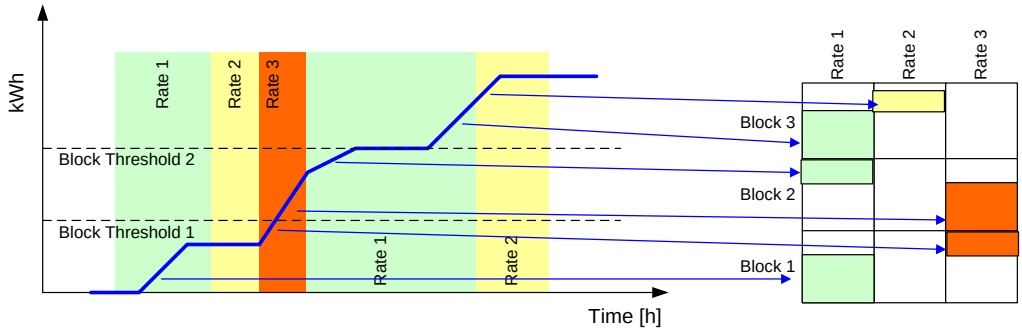


Figure D-56– Single Threshold Set applied to All Consumption

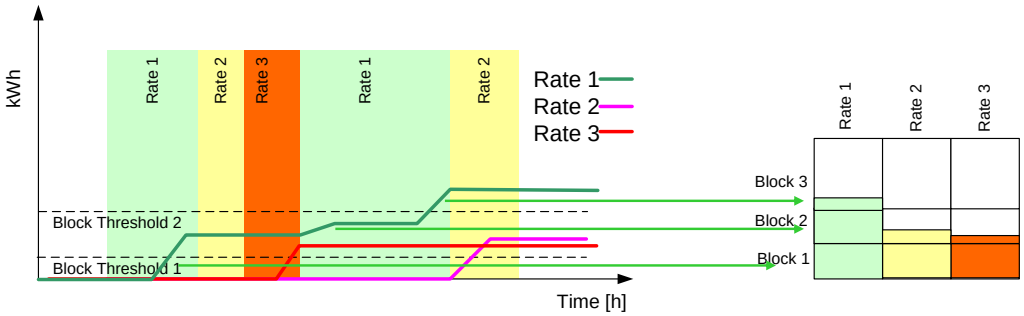


Figure D-57– Single Threshold Set applied to Each Tier Consumption

Note: Tiers 1-15 ONLY are available for hybrid Tier/Block tariffing ... Tiers 16-48 are for TOU tariffing only.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

8181 D.4.2.2.7.5 Unit of Measure Attribute

8182 An 8-bit enumeration identifying the base unit of measure. The enumeration used for this
8183 attribute shall match one of the *UnitOfMeasure* values using a pure Binary format, as defined in
8184 the Metering cluster.

8185 D.4.2.2.7.6 Currency Attribute

8186 An unsigned 16-bit integer containing identifying information concerning the local unit of
8187 currency used in the Price cluster. The *Currency* attribute shall correspond to the *Currency* field
8188 within the *PublishPrice* command.

8189 The value of the currency attribute should match the values defined by ISO 4217.

8190 D.4.2.2.7.7 PriceTrailingDigit Attribute

8191 An 8-bit BitMap used to determine where the decimal point is located for prices provided in the
8192 *Standing Charge* attribute and the *Price Matrix* command. The most significant nibble is the
8193 Trailing Digit sub-field which indicates the number of digits to the right of the decimal point.
8194 The least significant nibble is reserved and shall be 0. The *Price Trailing Digit* attribute
8195 represents the current active value.

8196 D.4.2.2.7.8 TariffResolutionPeriod Attribute

8197 An 8 bit enumeration identifying the resolution period for Block Tariff, Table D-107 shows all
8198 available options.

8199 D.4.2.2.7.9 CO_2 Attribute

8200 Used to calculate the amount of carbon dioxide (CO_2) produced from energy use. Natural gas has
8201 a conversion factor of about 0.185, e.g. 1,000 kWh of gas used is responsible for the production
8202 of 185kg CO_2 (0.185 x 1000 kWh). The *CO_2* attribute represents the current active value.

8203 D.4.2.2.7.10 CO_2Unit Attribute

8204 This attribute is an 8-bit enumeration which defines the unit for the CO_2 attribute. The values and
8205 descriptions for this attribute are listed in Table D-83 below. The *CO_2Unit* attribute represents
8206 the current active value.

8207

Table D-83– CO_2Unit Enumeration

Values	Description
0x00	Reserved for future use
0x01	kg per kWh
0x02	kg per Gallon of Gasoline
0x03	kg per Therm of Natural Gas
0x04 to 0xFF	Reserved for future use

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

8208

8209 D.4.2.2.7.11 CO_2TrailingDigit Attribute

8210 An 8-bit Bit-Map used to determine where the decimal point is located in the CO_2 attribute. The
8211 most significant nibble indicates the number of digits to the right of the decimal point. The least
8212 significant nibble is reserved. The *CO_2TrailingDigit* attribute represents the current active value.

8213 D.4.2.2.8 **Billing Information (Delivered) Attribute Set**

8214 The following set of attributes provides remote access to the Price server Billing information.

8215

8216 **Table D-84– Billing Information Attribute Set**

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory /Optional
0x0700	*CurrentBillingPeriodStart*	UTCTime	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x0701	*CurrentBillingPeriodDuration*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read only	-	O
0x0702	*LastBillingPeriodStart*	UTCTime	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x0703	*LastBillingPeriodDuration*	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read only	-	O
0x0704	LastBillingPeriodConsolidated Bill	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x0705-0x07FF	Reserved					

8217

8218 D.4.2.2.8.1 *CurrentBillingPeriodStart* Attribute

8219 The *CurrentBillingPeriodStart* attribute represents the start time of the current billing period.

8220 D.4.2.2.8.2 *CurrentBillingPeriodDuration* Attribute

8221 The *CurrentBillingPeriodDuration* attribute represents the current billing period duration in
8222 minutes.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

8223 D.4.2.2.8.3 LastBillingPeriodStart Attribute

8224 The *LastBillingPeriodStart* attribute represents the start time of the last billing period.

8225 D.4.2.2.8.4 LastBillingPeriodDuration Attribute

8226 The *LastBillingPeriodDuration* attribute is the duration of the last billing period in minutes (start
8227 to end of last billing period).

8228 D.4.2.2.8.5 LastBillingPeriodConsolidatedBill Attribute

8229 The *LastBillingPeriodConsolidatedBill* attribute is an amount for the cost of the energy supplied
8230 from the date of the *LastBillingPeriodStart* attribute and until the duration of the
8231 *LastBillingPeriodDuration* attribute expires, measured in base unit of *Currency* with the decimal
8232 point located as indicated by the Trailing Digits attribute.

8233 D.4.2.2.9 **Credit Payment Attribute Set**

8234 The Credit Payments Attribute set provides a method for the HAN (IHD) to understand the
8235 current status of the credit-only payment made to the energy supplier. These payments are for a
8236 credit meter only and do not cover any Prepayment Top up or payment. This attribute set is used
8237 to display the bill on the IHD should this service be required. Devices that require this
8238 information should use standard ZCL commands to read this information.

8239 **Table D-85– Credit Payment Attribute Set**

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0800	CreditPaymentDueDate	UTC Time		Read Only	-	O
0x0801	CreditPaymentStatus	8 bit Enumeration	0x00 – 0xFF	Read Only	-	O
0x0802	CreditPayment OverDueAmount	Signed 32-bit Integer	- 0x7FFFFFFF - + 0x7FFFFFF	Read Only	0	O
0x0803 – 0x0809	Reserved					
0x080A	PaymentDiscount	Signed 32-bit Integer	- 0x7FFFFFFF - + 0x7FFFFFF	Read Only	-	O
0x080B	PaymentDiscount Period	8 bit Enumeration	0x00 – 0xFF	Read Only	-	O
0x0801 – 0x080F	Reserved					
0x0810	CreditPayment#1	Unsigned 32bit Integer	0x00000000 – 0xFFFFFFFF	Read Only	-	O
0x0811	CreditPaymentDate#1	UTC Time		Read Only	-	O
0x0812	CreditPaymentRef#1	Octet String	1-21	Read Only	-	O
0x0813 – 0x081F	Reserved					

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0820	CreditPayment#2	Unsigned 32bit Integer	0x00000000 – 0xFFFFFFFF	Read Only	-	O
0x0821	CreditPaymentDate#2	UTC Time		Read Only	-	O
0x0822	CreditPaymentRef#2	Octet String	1-21	Read Only	-	O
0x0823 – 0x082F	Reserved					
0x0830	CreditPayment#3	Unsigned 32bit Integer	0x00000000 – 0xFFFFFFFF	Read Only	-	O
0x0831	CreditPaymentDate#3	UTC Time		Read Only	-	O
0x0832	CreditPaymentRef#3	Octet String	1-21	Read Only	-	O
0x0833 – 0x083F	Reserved					
0x0840	CreditPayment#4	Unsigned 32bit Integer	0x00000000 – 0xFFFFFFFF	Read Only	-	O
0x0841	CreditPaymentDate#4	UTC Time		Read Only	-	O
0x0842	CreditPaymentRef#4	Octet String	1-21	Read Only	-	O
0x0843 – 0x084F	Reserved					
0x0850	CreditPayment#5	Unsigned 32bit Integer	0x00000000 – 0xFFFFFFFF	Read Only	-	O
0x0851	CreditPaymentDate#5	UTC Time		Read Only	-	O
0x0852	CreditPaymentRef#5	Octet String	1-21	Read Only	-	O
0x0853 – 0x08FF	Reserved					

8240

8241 D.4.2.2.9.1 CreditPaymentdueDate Attribute

8242 The *CreditPaymentDueDate* attribute indicates the date and time when the next credit payment is
8243 due to be paid by the consumer to the supplier.

8244 D.4.2.2.9.2 CreditPaymentStatus Attribute

8245 The *CreditPaymentStatus* attribute indicates the current status of the last payment. Table D-86
8246 defines the enumeration values for this attribute.

8247

Table D-86– CreditPaymentStatus Enumeration

Enumerated Value	Status
0x00	Pending
0x01	Received / Paid
0x02	Overdue
0x03	2 payments overdue
0x04	3 payments overdue

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x05 – 0xFF	Reserved

8248

8249 D.4.2.2.9.3 CreditPaymentOverDueAmount Attribute

8250 This is the total of the consolidated bill amounts accumulated since the last payment.

8251 D.4.2.2.9.4 PaymentDiscount Attribute

8252 The *PaymentDiscount* attribute indicates the discount that the energy supplier has applied to the
8253 consolidated bill.

8254 D.4.2.2.9.5 PaymentDiscountPeriod Attribute

8255 The *PaymentDiscountPeriod* attribute indicates the period for which this discount shall be
8256 applied for. Table D-87 shows the enumeration values for this attribute.

8257

Table D-87– PaymentDiscountDuration Enumerations

Enumerated Value	Status
0x00	Current Billing Period
0x01	Current Consolidated bill
0x02	One Month
0x03	One Quarter
0x04	One Year
0x05 – 0xFF	reserved

8258

8259 D.4.2.2.9.6 CreditPayment Attribute

8260 The *CreditPayment* attributes indicate the amount paid by the consumer to the energy supplier.
8261 The last 5 values are shown with #1 meaning the most recent. Measured in base unit of *Currency*
8262 with the decimal point located as indicated by the *Trailing Digits* attribute.

8263 D.4.2.2.9.7 CreditPaymentDate Attribute

8264 The *CreditPaymentDate* attributes indicate the last time the consumer made a payment to the
8265 energy supplier. The last 5 values are shown with #1 meaning the most recent.

8266 D.4.2.2.9.8 CreditPaymentRef Attribute

8267 The *CreditPaymentRef* attributes indicate the reference number given to the payment by the
8268 energy supplier. The last 5 values are shown with #1 meaning the most recent.

8269 D.4.2.2.10 **Received Tier Label Attribute Set**

8270

8271 **Table D-88– Received Tier Label Attribute Set**

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x8000	ReceivedTier1 PriceLabel	Octet string	1 to 13	Read/ Write	"Tier 1"	O
0x800n	ReceivedTierN PriceLabel	Octet string	1 to 13	Read/ Write	"Tier N"	O
0x802F	ReceivedTier48 PriceLabel	Octet string	1 to 13	Read/ Write	"Tier 48"	O
0x8030-0x80FF	Reserved					

8272

8273 D.4.2.2.10.1 ReceivedTierNPriceLabel Attributes

8274 The *ReceivedTierNPriceLabel* attributes provide a method for utilities to assign a label to
8275 Received Price Tiers. There are 48 Tier Labels. The format and use of these attributes is the same
8276 as for the 'Delivered' Price Labels defined in D.4.2.2.1.1.

8277 D.4.2.2.11 **Received Block Threshold Attribute Set**

8278 The following set of attributes provides remote access to the Price server
8279 *ReceivedBlockThresholds*. The number of block thresholds is indicated by the
8280 *NumberofBlockThresholds* field in the associated *PublishTariffInformation* command. The
8281 number of blocks is one greater than the number of thresholds.

8282 **Table D-89– Received Block Threshold Attribute Set**

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x8100	ReceivedBlock1Threshold	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x810n	ReceivedBlockNThreshold	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x810E	ReceivedBlock15Threshold	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x810F-0x81FF	Reserved					

8283

8284 D.4.2.2.11.1 ReceivedBlockNThreshold Attributes

8285 The format of these attributes is the same as for the 'Delivered' Block Thresholds defined in
8286 D.4.2.2.2.1.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

8287 D.4.2.2.12 **Received Block Period Attribute Set**

8288 The following set of attributes provides remote access to the Price server Received Block
8289 Threshold period (typically the billing cycle) information.

8290

8291 **Table D-90– Received Block Period Attribute Set**

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x8200	ReceivedStartofBlockPeriod	UTC Time	-	Read Only	-	O
0x8201	ReceivedBlockPeriodDuration	Unsigned 24 bit Integer	0x000000 – 0xFFFFFF	Read Only	-	O
0x8202	ReceivedThresholdMultiplier	Unsigned 24 bit Integer	0x000000 – 0xFFFFFF	Read Only	-	O
0x8203	ReceivedThresholdDivisor	Unsigned 24 bit Integer	0x000000 – 0xFFFFFF	Read Only	-	O
0x8204 – 0x82FF	Reserved					

8292

8293 D.4.2.2.12.1 ReceivedStartofBlockPeriod Attribute

8294 The format of this attribute is the same as for the 'Delivered' *StartofBlockPeriod* attribute
8295 defined in D.4.2.2.3.1.

8296 D.4.2.2.12.2 ReceivedBlockPeriodDuration Attribute

8297 The format of this attribute is the same as for the 'Delivered' *BlockPeriodDuration* attribute
8298 defined in D.4.2.2.3.2.

8299 D.4.2.2.12.3 ReceivedThresholdMultiplier Attribute

8300 The format of this attribute is the same as for the 'Delivered' *ThresholdMultiplier* attribute
8301 defined in D.4.2.2.3.3.

8302 D.4.2.2.12.4ReceivedThresholdDivisor Attribute

8303 The format of this attribute is the same as for the 'Delivered' *ThresholdDivisor* attribute
8304 defined in D.4.2.2.3.4.

8305 D.4.2.2.13 **Received Block Price Information Attribute Set**

8306 **Table D-91– Received Block Price Attribute Set**

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x8400	RxNoTierBlock1Price	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x8401	RxNoTierBlock2Price	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x8402	RxNoTierBlock3Price	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x840N	RxNoTierBlockN+1Price …	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x840F	RxNoTierBlock16Price	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x8410	RxTier1Block1Price	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
…	…	…	…	…	…	…
0x84FF	RxTier15Block16Price	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read only	-	O

8307

8308 D.4.2.2.13.1 RxTierNBlockNPrice Attributes

8309 The format and use of these attributes is the same as for the 'Delivered' *TierNBlockNPrice*
8310 attributes defined in D.4.2.2.5.1.

8311 D.4.2.2.14 **Received Extended Price Information Attribute Set**

8312 In case of TOU charging only, the price server shall support up to 48 TOU rates. To reduce the
8313 number of attributes, *RxTierNBlock1Price* attributes are reused to represent rates for tiers 1 to 15.
8314 Rates for tiers 16 to 48 are provided in the Received Extended Price Information Set.

8315 **Table D-92– Received Extended Price Information Attribute Set (TOU charging only)**

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x8500-0x850E	Reserved					
0x850F	ReceivedPriceTier16	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x8510	ReceivedPriceTier17	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x8511	ReceivedPriceTier18	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
…	ReceivedPriceTierN	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x852F	ReceivedPriceTier48	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x8530-0x85FF	Reserved					

8316 D.4.2.2.14.1 ReceivedPriceTierN Attributes

8317 The format and use of these attributes is the same as for the 'Delivered' *PriceTierN* attributes
8318 defined in D.4.2.2.6.1.

8319 D.4.2.2.15 **Received Tariff Information Attribute Set**

8320 The following set of attributes represents items that are associated with a particular Received
8321 Price Tariff.

8322 **Table D-93– Received Tariff Information Attribute Set**

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x8600-0x860F	Reserved					
0x8610	ReceivedTariffLabel	Octet String	1 to 25 Octets	Read Only	0	O
0x8611	ReceivedNumberof PriceTiersInUse	Unsigned 8-bit Integer	0 to 15	Read Only	0	O
0x8612	ReceivedNumberof BlockThresholdsInUse	Unsigned 8-bit Integer	0 to 15	Read Only	0	O
0x8613	ReceivedTierBlock Mode	Unsigned 8-bit Integer	0 to 1	Read Only	0xFF	O
0x8614	Reserved					
0x8615	ReceivedTariff ResolutionPeriod	8-bit Enumeration	0x00 to 0xFF	Read Only	0x00	O
0x8616-0x8624	Reserved					
0x8625	ReceivedCO_2	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	185	O
0x8626	ReceivedCO_2Unit	8-bit Enumeration		Read Only	1	O
0x8627	ReceivedCO_2 TrailingDigit	8-bit BitMap		Read Only	0	O
0x8628-0x86FF	Reserved					

8323

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

8324 D.4.2.2.15.1 ReceivedTariffLabel Attribute

8325 The format and use of this attribute is the same as for the 'Delivered' *TariffLabel* attribute
8326 defined in D.4.2.2.7.1.

8327 D.4.2.2.15.2 ReceivedNumberofPriceTiersInUse Attribute

8328 The format and use of this attribute is the same as for the 'Delivered' *NumberofPriceTiersInUse*
8329 attribute defined in D.4.2.2.7.2.

8330 D.4.2.2.15.3 ReceivedNumberofBlockThresholdsInUse Attribute

8331 The format and use of this attribute is the same as for the 'Delivered'
8332 *NumberofBlockThresholdsInUse* attribute defined in D.4.2.2.7.3.

8333 D.4.2.2.15.4 ReceivedTierBlockMode Attribute

8334 The format and use of this attribute is the same as for the 'Delivered' *TierBlockMode* attribute
8335 defined in D.4.2.2.7.4.

8336 D.4.2.2.15.5 ReceivedTariffResolutionPeriod Attribute

8337 An 8 bit enumeration identifying the resolution period for Block Tariff, Table D-107 shows all
8338 available options.

8339 D.4.2.2.15.6 ReceivedCO_2 Attribute

8340 The format and use of this attribute is the same as for the 'Delivered' *CO_2* attribute defined in
8341 D.4.2.2.7.9.

8342 D.4.2.2.15.7 ReceivedCO_2Unit Attribute

8343 The format and use of this attribute is the same as for the 'Delivered' *CO_2Unit* attribute defined
8344 in D.4.2.2.7.10.

8345 D.4.2.2.15.8 ReceivedCO_2TrailingDigit Attribute

8346 The format and use of this attribute is the same as for the 'Delivered' *CO_2TrailingDigit* attribute
8347 defined in D.4.2.2.7.11.

8348 D.4.2.2.16 **Received Billing Information Attribute Set**

8349 The following set of attributes represents items that are associated with particular Received
8350 Billing information.

8351 **Table D-94– Received Billing Information Attribute Set**

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x8700	ReceivedCurrentBilling PeriodStart	UTC Time	0x00000000 to 0xFFFFFFFF	Read Only	-	O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x8701	ReceivedCurrentBilling Period Duration	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x8702	ReceivedLastBilling PeriodStart	UTC Time	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x8703	ReceivedLastBilling PeriodDuration	Unsigned 24-bit Integer	0x000000 to 0xFFFFFF	Read Only	-	O
0x8704	ReceivedLastBilling Period ConsolidatedBill	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x8705-0x87FF	Reserved					

8352

8353 D.4.2.2.16.1 ReceivedCurrentBillingPeriodStart Attribute

8354 The format and use of this attribute is the same as for the ‘Delivered’ *CurrentBillingPeriodStart*
8355 attribute defined in D.4.2.2.8.1.

8356 D.4.2.2.16.2 ReceivedCurrentBillingPeriodDuration Attribute

8357 The format and use of this attribute is the same as for the ‘Delivered’
8358 *CurrentBillingPeriodDuration* attribute defined in D.4.2.2.8.2.

8359 D.4.2.2.16.3 ReceivedLastBillingPeriodStart Attribute

8360 The format and use of this attribute is the same as for the ‘Delivered’ *LastBillingPeriodStart*
8361 attribute defined in D.4.2.2.8.3.

8362 D.4.2.2.16.4 ReceivedLastBillingPeriodDuration Attribute

8363 The format and use of this attribute is the same as for the ‘Delivered’ *LastBillingPeriodDuration*
8364 attribute defined in D.4.2.2.8.4.

8365 D.4.2.2.16.5 ReceivedLastBillingPeriodConsolidatedBill Attribute

8366 The format and use of this attribute is the same as for the ‘Delivered’
8367 *LastBillingPeriodConsolidatedBill* attribute defined in D.4.2.2.8.5.

8368

8369 ### D.4.2.3 Commands Received

8370 The server side of the Price cluster is capable of receiving the commands listed in Table D-95.

8371 **Table D-95– Received Command IDs for the Price Cluster**

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Command Identifier Field Value	Description	Mandatory / Optional
0x00	*GetCurrentPrice*	M
0x01	*GetScheduledPrices*	O
0x02	*PriceAcknowledgement*	M - Mandatory for 1.1 and later devices
0x03	*GetBlockPeriod(s)*	O
0x04	*GetConversionFactor*	O
0x05	*GetCalorificValue*	O
0x06	*GetTariffInformation*	O
0x07	*GetPriceMatrix*	O
0x08	*GetBlockThresholds*	O
0x09	*GetCO_2Value*	O
0x0A	*GetTierLabels*	O
0x0B	*GetBillingPeriod*	O
0x0C	*GetConsolidatedBill*	O
0x0D	*CPPEventResponse*	O
0x0E	*GetCreditPayment*	O
0x0F	*GetCurrencyConversion*	O
0x10	*GetTariffCancellation*	O
0x11 to 0xFF	Reserved	

8372

8373 D.4.2.3.1 **Error Handling**

8374 If the response to a 'Get' command has no data available, then the device should respond using a
8375 ZCL Default Response with a status of NOT_FOUND.

8376 D.4.2.3.2 **GetCurrentPrice Command**

8377 This command initiates a *PublishPrice* command (see sub-clause D.4.2.4.1) for the current time.

8378 D.4.2.3.2.1 Payload Format

8379 The payload of the *GetCurrentPrice* command is formatted as shown in Figure D-58:

Octets	1
Data Type	Unsigned 8-bit integer
Field Name	Command Options

8380 **Figure D-58– *GetCurrentPrice* Command Payload**

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

8381

8382 ***D.4.2.3.2.1.1 Payload Details***

8383 **The Command Options Field:** The command options field is 8 Bits in length and is formatted as
8384 a bit field as shown in Figure D-59.

8385

Bits	0	1 to 7
Field Name	Requestor Rx On When Idle	Reserved

8386 **Figure D-59– *Get Current Price* Command Options Field**

8387 **The Requestor Rx On When Idle Sub-field:** The Requestor Rx On When Idle sub-field has a
8388 value of 1 if the requestor's receiver may be, for all practical purposes, enabled when the
8389 device is not actively transmitting, thereby making it very likely that regular broadcasts of
8390 pricing information will be received by this device, and 0 otherwise.

8391 A device that publishes price information may use the value of this bit, as received from requestors
8392 in its neighborhood, to determine publishing policy. For example, if a device makes a request for
8393 current pricing information and the requestor Rx on when idle sub-field of the *GetCurrentPrice*
8394 command payload has a value of 1 (indicating that the device will be likely to receive regular
8395 price messages), then the receiving device may store information about the requestor and use it in
8396 future publishing operations.

8397 D.4.2.3.2.2 Effect on Receipt

8398 On receipt of this command, the device shall send a *PublishPrice* command (sub-clause
8399 D.4.2.4.1) for the currently scheduled time.

8400

8401 D.4.2.3.3 **GetScheduledPrices Command**

8402 This command initiates a *PublishPrice* command (see sub-clause D.4.2.4.1) for available price
8403 events. A server device shall be capable of storing five price events at a minimum.

8404 D.4.2.3.3.1 Payload Details

8405 The *GetScheduledPrices* command payload shall be formatted as illustrated in Figure D-60:

8406

Octets	**4**	**1**

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Data Type	UTCTime	Unsigned8-bit integer
Field Name	StartTime (M)	NumberofEvents (M)

8407 **Figure D-60– *GetScheduledPrices* Command Payload**

8408 **StartTime (mandatory):** UTC Timestamp representing the minimum ending time for any
8409 scheduled or currently active pricing events to be resent. If a command has a *StartTime* of
8410 0x00000000, replace that *StartTime* with the current time stamp.

8411 **NumberofEvents (mandatory):** Represents the maximum number of events to be sent. A value
8412 of 0 would indicate all available events are to be returned. Example: Number of Events = 1
8413 would return the first event with an EndTime greater than or equal to the value of *StartTime*
8414 field in the *GetScheduledPrices* command. (EndTime would be *StartTime* plus Duration of the
8415 event listed in the device's event table).

8416 D.4.2.3.3.2 When Generated

8417 This command is generated when the client device wishes to verify the available Price Events or
8418 after a loss of power/reset occurs and the client device needs to recover currently active,
8419 scheduled, or expired Price Events.

8420 A ZCL Default Response with status NOT_FOUND shall be returned if there are no events
8421 available.

8422 D.4.2.3.3.3 Effect on Receipt

8423 On receipt of this command, the device shall send a *PublishPrice* command (see sub-clause
8424 D.4.2.4.1) for all currently scheduled price events.

8425 D.4.2.3.4 **PriceAcknowledgement Command**

8426 The *PriceAcknowledgement* command described in Figure D-61 provides the ability to
8427 acknowledge a previously sent *PublishPrice* command. It is mandatory for 1.1 and later devices.
8428 For SE 1.0 devices, the command is optional.

8429 D.4.2.3.4.1 Payload Format

Octets	4	4	4	1
Data Type	Unsigned 32 bit Integer	Unsigned 32 bit Integer	UTCTime	8 bit BitMap
Field Name	Provider ID (M)	Issuer Event ID (M)	Price Ack Time (M)	Control (M)

8430 **Figure D-61– *PriceAcknowledgement* Command Payload**

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

8431 ***D.4.2.3.4.1.1 Payload Details***

8432 **Provider ID (mandatory):** An unsigned 32 bit field containing a unique identifier for
8433 the commodity provider.

8434 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider.

8435 **Price Ack Time (mandatory):** Time price acknowledgement generated.

8436 **Control (mandatory):** Identifies the Price Control or Block Period Control options for the
8437 event. The values for this field are described in Table D-100 and Figure D-75.

8438 D.4.2.3.4.2 When Generated

8439 This command is generated on receipt of a *PublishPrice* command when the PriceControl
8440 field of that *PublishPrice* command indicates that a Price Acknowledgement is required
8441 (see sub-clause D.4.2.4.1 for further details).

8442

8443 D.4.2.3.5 **GetBlockPeriod(s) Command**

8444 This command initiates a *PublishBlockPeriod* command (see sub-clause D.4.2.4.2) for the
8445 currently scheduled block periods. A server device shall be capable of storing at least two
8446 commands, the current period and a period to be activated in the near future.

8447 D.4.2.3.5.1 Payload Format

Octets	4	1	1
Data Type	UTCTime	Unsigned 8 bit Integer	8-bit Bitmap
Field Name	Start Time (M)	Number of Events (M)	Tariff Type (O)

8448 **Figure D-62– *Get Block Period(s)* Command Payload**

8449 ***D.4.2.3.5.1.1 Payload Details***

8450 **Start Time (mandatory):** UTC Timestamp representing the minimum ending time for any
8451 scheduled or currently block period events to be resent. If a command has a Start Time of
8452 0x00000000, replace that Start Time with the current time stamp.

8453 **Number of Events (mandatory):** An 8 bit Integer which indicates the maximum number of
8454 *Publish Block Period* commands that can be sent. Example: Number of Events = 1 would return
8455 the first event with an EndTime greater than or equal to the value of Start Time field in the
8456 *GetBlockPeriod(s)* command. (EndTime would be StartTime plus Duration of the event listed in

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

8457 the device's event table). Number of Events = 0 would return all available Publish Block
8458 Periods, starting with the current block in progress.

8459 **Tariff Type (mandatory):** An 8-bit bitmap identifying the type of tariff published in this
8460 command. The least significant nibble represents an enumeration of the tariff type as detailed in
8461 Table D-108 (Generation Meters shall use the 'Received' Tariff.). If the *TariffType* is not
8462 specified, the server shall assume that the request is for the 'Delivered' Tariff. The most
8463 significant nibble is reserved.

8464 D.4.2.3.5.2 When Generated

8465 This command is generated when the client device wishes to verify the available Block Period
8466 events or after a loss of power/reset occurs and the client device needs to recover currently
8467 active or scheduled Block Periods.

8468 A ZCL Default response with status NOT_FOUND shall be returned if there are no events
8469 available.

8470 D.4.2.3.5.3 Effect on Receipt

8471 On receipt of this command, the device shall send a *Publish Block Period* command
8472 (sub-clause D.4.2.4.2) for all currently scheduled periods, up to the maximum number of
8473 commands specified.

8474

8475 D.4.2.3.6 **GetConversionFactor Command**

8476 This command initiates a *PublishConversionFactor* command(s) for scheduled conversion factor
8477 updates. A server device shall be capable of storing at least two instances, the current and (if
8478 available) next instance to be activated in the future.

8479 D.4.2.3.6.1 Payload Format

Octets	4	4	1
Data Type	UTC Time	Unsigned 32-bit Integer	Unsigned 8-bit Integer
Field Name	Earliest Start Time (M)	Min. Issuer Event ID (M)	Number of Commands (M)

8480 **Figure D-63– *GetConversionFactor* Command Payload**

8481 D.4.2.3.6.2 Payload Details

8482 **Earliest Start Time (mandatory):** UTC Timestamp indicating the earliest start time of values to
8483 be returned by the corresponding *PublishConversionFactor* command. The first returned
8484 *PublishConversionFactor* command shall be the instance which is active or becomes active at

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

or after the stated *Earliest Start Time*. If more than one instance is requested, the active and scheduled instances shall be sent with ascending ordered StartTime.

Min. Issuer Event ID (mandatory): A 32-bit integer representing the minimum *Issuer Event ID* of values to be returned by the corresponding *PublishCalorificValue* command. A value of 0xFFFFFFFF means not specified; the server shall return values irrespective of the value of the *Issuer Event ID*.

Number of Commands (mandatory): An 8-bit integer which represents the maximum number of *PublishConversionFactor* commands that the client is willing to receive in response to this command. A value of 0 would indicate all available *PublishConversionFactor* commands shall be returned.

D.4.2.3.6.3 Effect on Receipt

A ZCL Default response with status NOT_FOUND shall be returned if there are no conversion factor updates available.

D.4.2.3.7 GetCalorificValue Command

This command initiates a *PublishCalorificValue* command(s) for scheduled calorific value updates. A server device shall be capable of storing at least two instances, the current and (if available) next instance to be activated in the future.

D.4.2.3.7.1 Payload Format

Octets	4	4	1
Data Type	UTC Time	Unsigned 32-bit Integer	Unsigned 8-bit Integer
Field Name	Earliest Start Time (M)	Min. Issuer Event ID (M)	Number of Commands (M)

Figure D-64– *GetCalorificValue* Command Payload

D.4.2.3.7.2 Payload Details

Earliest Start Time (mandatory): UTC Timestamp indicating the earliest start time of values to be returned by the corresponding *PublishCalorificValue* command. The first returned *PublishCalorificValue* command shall be the instance which is active or becomes active at or after the stated *Earliest Start Time*. If more than one instance is requested, the active and scheduled instances shall be sent with ascending ordered Start Time.

Min. Issuer Event ID (mandatory): A 32-bit integer representing the minimum *Issuer Event ID* of values to be returned by the corresponding *PublishCalorificValue* command. A value of

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

8513 0xFFFFFFFF means not specified; the server shall return values irrespective of the value of the
8514 *Issuer Event ID*.

8515 **Number of Commands (mandatory):** An 8-bit Integer which represents the maximum number
8516 of *PublishCalorificValue* commands that the client is willing to receive in response to this
8517 command. A value of 0 would indicate all available *PublishCalorificValue* commands shall be
8518 returned.

8519 D.4.2.3.7.3 Effect on Receipt

8520 A ZCL Default response with status NOT_FOUND shall be returned if there are no calorific
8521 value updates available.

8522

8523 D.4.2.3.8 **GetTariffInformation Command**

8524 This command initiates *PublishTariffInformation* command(s) for scheduled tariff updates. A
8525 server device shall be capable of storing at least **two** instances, current and the next instance to be
8526 activated in the future.

8527 One or more *PublishTariffInformation* commands are sent in response to this command.

8528 To obtain the complete tariff details, further *GetPriceMatrix* and *GetBlockThesholds* commands
8529 must be sent using the start time and *IssuerTariffID* obtained from the appropriate
8530 *PublishTariffInformation* command.

8531 D.4.2.3.8.1 Payload Format

Octets	**4**	**4**	**1**	**1**
Data Type	UTC Time	Unsigned 32-bit Integer	Unsigned 8-bit Integer	8-bit Bitmap
Field Name	Earliest Start Time (M)	Min. Issuer Event ID (M)	Number of Commands (M)	Tariff Type (M)

8532 **Figure D-65– GetTariffInformation Command Payload**

8533

8534 D.4.2.3.8.2 Payload Details

8535 **Earliest Start Time (mandatory):** UTC Timestamp indicating the earliest start time of tariffs to
8536 be returned by the corresponding *PublishTariffInformation* command. The first returned
8537 *PublishTariffInformation* command shall be the instance which is active or becomes active at or
8538 after the stated *EarliestStartTime*. If more than one command is requested, the active and
8539 scheduled commands shall be sent with ascending ordered *StartTime*.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Min. Issuer Event ID (mandatory): A 32-bit integer representing the minimum *Issuer Event ID* of tariffs to be returned by the corresponding *PublishTariffInformation* command. A value of 0xFFFFFFFF means not specified; the server shall return tariffs irrespective of the value of the *Issuer Event ID*.

Number of Commands (mandatory): An 8-bit integer which represents the maximum number of *PublishTariffInformation* commands that the client is willing to receive in response to this command. A value of 0 would indicate all available *PublishTariffInformation* commands shall be returned.

Tariff Type (mandatory): An 8-bit bitmap identifying the type of tariff published in this command. The least significant nibble represents an enumeration of the tariff type as detailed in Table D-108 (Generation Meters shall use the 'Received' Tariff.). The most significant nibble is reserved.

D.4.2.3.8.3 Effect on Receipt

A ZCL Default response with status NOT_FOUND shall be returned if there are no tariff updates available.

D.4.2.3.9 GetPriceMatrix Command

This command initiates a *PublishPriceMatrix* command for the scheduled Price Matrix updates. A server device shall be capable of storing at least **two** instances, current and next instance to be activated in the future.

D.4.2.3.9.1 Payload Format

Octets	**4**
Data Type	Unsigned 32-bit Integer
Field Name	Issuer Tariff ID (M)

Figure D-66– GetPriceMatrix Command Payload

D.4.2.3.9.2 Payload Details

Issuer Tariff ID (mandatory): *IssuerTariffID* indicates the tariff to which the requested Price Matrix belongs.

Note: A Price Matrix instance may require multiple *PublishPriceMatrix* commands to be transmitted to the client device.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

8567 D.4.2.3.9.3 Effect on Receipt

8568 A ZCL Default response with status NOT_FOUND shall be returned if there are no Price
8569 Matrix updates available.

8570

8571 D.4.2.3.10 **GetBlockThresholds Command**

8572 This command initiates a *PublishBlockThreshold* command for the scheduled Block Threshold
8573 updates. A server device shall be capable of storing at least **two** instances, current and next
8574 instance to be activated in the future.

8575 D.4.2.3.10.1 Payload Format

Octets	**4**
Data Type	Unsigned 32-bit Integer
Field Name	Issuer Tariff ID (M)

8576 **Figure D-67– GetBlockThresholds Command Payload**

8577 D.4.2.3.10.2 Payload Details

8578 **Issuer Tariff ID (mandatory):** Issuer Tariff ID indicates the tariff to which the requested Block
8579 Thresholds belong.

8580 **Note:** A Block Threshold instance may require multiple *PublishBlockThreshold* commands to be
8581 transmitted to the client device.

8582 D.4.2.3.10.3 Effect on Receipt

8583 A ZCL Default response with status NOT_FOUND shall be returned if there are no Block
8584 Threshold updates available.

8585

8586 D.4.2.3.11 **GetCO_2Value Command**

8587 This command initiates *PublishCO_2Value* command(s) for scheduled CO_2 conversion factor
8588 updates. A server device shall be capable of storing at least **two** instances, current and (if
8589 available) next instance to be activated in the future.

8590 D.4.2.3.11.1 Payload Format

Octets	**4**	**4**	**1**	**1**
Data Type	UTC Time	Unsigned 32-bit Integer	Unsigned 8-bit Integer	8-bit Bitmap
Field Name	Earliest Start Time (M)	Min. Issuer Event ID (M)	Number of Commands (M)	Tariff Type (O)

8591 **Figure D-68– GetCO_2Value Command Payload**

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

8592 D.4.2.3.11.2 Payload Details

8593 **Earliest Start Time (mandatory):** UTC Timestamp indicating the earliest start time of values to
8594 be returned by the corresponding *PublishCO$_2$Value* command. The first returned
8595 *PublishCO$_2$Value* command shall be the instance which is active or becomes active at or after the
8596 stated *EarliestStartTime*. If more than one instance is requested, the active and scheduled
8597 instances shall be sent with ascending ordered *StartTime*.

8598 **Min. Issuer Event ID (mandatory):** A 32-bit integer representing the minimum *Issuer Event ID*
8599 of values to be returned by the corresponding *PublishCO$_2$Value* command. A value of
8600 0xFFFFFFFF means not specified; the server shall return values irrespective of the value of the
8601 *Issuer Event ID*.

8602 **Number of Commands (mandatory):** An 8-bit Integer which represents the maximum number
8603 of *PublishCO$_2$Value* commands that the client is willing to receive in response to this command.
8604 A value of 0 would indicate all available *PublishCO$_2$Value* commands shall be returned.

8605 **Tariff Type (Optional):** An 8-bit bitmap identifying the type of tariff published in this
8606 command. The least significant nibble represents an enumeration of the tariff type as detailed in
8607 Table D-108 (Generation Meters shall use the 'Received' Tariff). A value of 0xFF means not
8608 specified. If the *TariffType* is not specified, the server shall return all $C0_2$ values regardless of
8609 tariff type. The most significant nibble is reserved.

8610 D.4.2.3.11.3 Effect on Receipt

8611 A ZCL Default response with status NOT_FOUND shall be returned if there are no CO_2
8612 conversion factor updates available.

8613

8614 D.4.2.3.12 **GetTierLabels Command**

8615 This command allows a client to retrieve the tier labels associated with a given tariff; this
8616 command initiates a *PublishTierLabels* command from the server.

8617 D.4.2.3.12.1 Payload Format

Octets	**4**
Data Type	Unsigned 32-bit Integer
Field Name	Issuer Tariff ID (M)

8618 **Figure D-69– GetTierLabels Command Payload**

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

8619 D.4.2.3.12.2 Payload Details

8620 **Issuer Tariff ID (mandatory):** Unique identifier generated by the commodity supplier. This is
8621 used to identify the tariff that the labels apply to.

8622 D.4.2.3.12.3 Effect on Receipt

8623 A ZCL Default response with status NOT_FOUND shall be returned if there are no tier label
8624 updates available.

8625

8626 D.4.2.3.13 **GetBillingPeriod Command**

8627 This command initiates one or more *PublishBillingPeriod* commands for currently scheduled
8628 billing periods.

8629 D.4.2.3.13.1 Payload Format

Octets	**4**	**4**	**1**	**1**
Data Type	UTC Time	Unsigned 32-bit Integer	Unsigned 8-bit Integer	8-bit Bitmap
Field Name	Earliest Start Time (M)	Min. Issuer Event ID (M)	Number of Commands (M)	Tariff Type (O)

8630 **Figure D-70– GetBillingPeriod Command Payload**

8631 D.4.2.3.13.2 Payload Details

8632 **Earliest Start Time (mandatory):** UTC Timestamp indicating the earliest start time of billing
8633 periods to be returned by the corresponding *PublishBillingPeriod* command. The first returned
8634 *PublishBillingPeriod* command shall be the instance which is active or becomes active at or after
8635 the stated *EarliestStartTime*. If more than one instance is requested, the active and scheduled
8636 instances shall be sent with ascending ordered *StartTime*.

8637 **Min. Issuer Event ID (mandatory):** A 32-bit integer representing the minimum *Issuer Event ID*
8638 of billing periods to be returned by the corresponding *PublishBillingPeriod* command. A value of
8639 0xFFFFFFFF means not specified; the server shall return periods irrespective of the value of the
8640 *Issuer Event ID*.

8641 **Number of Commands (mandatory):** An 8 bit Integer which indicates the maximum number of
8642 *PublishBillingPeriod* commands that the client is willing to receive in response to this command.
8643 A value of 0 would indicate all available *PublishBillingPeriod* commands shall be returned.

8644 **Tariff Type (optional):** An 8-bit bitmap identifying the *TariffType* of the requested Billing
8645 Period information. The least significant nibble represents an enumeration of the tariff type as
8646 detailed in Table D-108 (Generation Meters shall use the 'Received' Tariff). A value of 0xFF
8647 means not specified. If the *TariffType* is not specified, the server shall return Billing Period
8648 information regardless of its type. The most significant nibble is reserved.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

8649 D.4.2.3.13.3 Effect on Receipt

8650 A ZCL Default response with status NOT_FOUND shall be returned if there are no scheduled
8651 billing periods available.

8652

8653 D.4.2.3.14 **GetConsolidatedBill Command**

8654 This command initiates one or more *PublishConsolidatedBill* commands with the requested
8655 billing information.

8656 D.4.2.3.14.1 Payload Format

Octets	**4**	**4**	**1**	**1**
Data Type	UTC Time	Unsigned 32-bit Integer	Unsigned 8-bit Integer	8-bit Bitmap
Field Name	Earliest Start Time (M)	Min. Issuer Event ID (M)	Number of Commands (M)	Tariff Type (O)

8657 **Figure D-71– GetConsolidatedBill Command Payload**

8658 D.4.2.3.14.2 Payload Details

8659 **Earliest Start Time (mandatory):** UTC Timestamp indicating the earliest start time of billing
8660 information to be returned by the corresponding *PublishConsolidatedBill* command. The first
8661 returned *PublishConsolidatedBill* command shall be the instance which is active or becomes
8662 active at or after the stated *EarliestStartTime*. If more than one instance is requested, the active
8663 and scheduled instances shall be sent with ascending ordered *StartTime*.

8664 **Min. Issuer Event ID (mandatory):** A 32-bit integer representing the minimum *Issuer Event ID*
8665 of billing information to be returned by the corresponding *PublishConsolidatedBill* command. A
8666 value of 0xFFFFFFFF means not specified; the server shall return information irrespective of the
8667 value of the *Issuer Event ID*.

8668 **Number of Commands (mandatory):** An 8 bit Integer which indicates the maximum number of
8669 *PublishConsolidatedBill* commands that can be sent. A value of 0 would indicate all available
8670 *PublishConsolidatedBill* commands shall be returned.

8671 **Tariff Type (Optional):** An 8-bit bitmap identifying the type of tariff published in this
8672 command. The least significant nibble represents an enumeration of the tariff type as detailed in
8673 Table D-108 (Generation Meters shall use the 'Received' Tariff). A value of 0xFF means not
8674 specified. If the *TariffType* is not specified, the server shall return all billing information
8675 regardless of tariff type. The most significant nibble is reserved.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

8676 D.4.2.3.14.3 Effect on Receipt

8677 A ZCL Default response with status NOT_FOUND shall be returned if there is no billing
8678 information available.

8679

8680 D.4.2.3.15 **CPPEventResponse Command**

8681 ***Note:*** *The CPPEventResponse command in this revision of this specification is provisional and*
8682 *not certifiable. This feature may change before reaching certifiable status in a future revision of*
8683 *this specification.*

8684 The *CPPEventResponse* command is sent from a Client (IHD) to the ESI to notify it of a Critical
8685 Peak Pricing event authorization.

8686 D.4.2.3.15.1 Payload Format

Octets	**4**	**1**
Data Type	Unsigned 32-bit Integer	8-bit Enumeration
Field Name	Issuer Event ID (M)	CPP Auth (M)

8687 **Figure D-72– CPPEventResponse Command Payload**

8688 D.4.2.3.15.2 Payload Details

8689 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. When
8690 new information is provided that replaces older information for the same time period, this field
8691 allows devices to determine which information is newer. The value contained in this field is a
8692 unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)
8693 identifying when the Publish command was issued. Thus, newer information will have a value in
8694 the *Issuer Event ID* field that is larger than older information.

8695 **CPP Auth (mandatory):** An 8-bit enumeration identifying the status of the CPP event. This
8696 field shall contain the 'Accepted' or 'Rejected' values defined in Table D-113.

8697 D.4.2.3.15.3 When Generated

8698 The *CPPEventResponse* command is sent in response to the *PublishCPPEvent* command, for
8699 either the Meter or the IHD, as acceptance or rejection of the CPP event.

8700 D.4.2.3.15.4 Effect on Receipt

8701 When the *CPPEventResponse* is received by the ESI, it will look at the *CPPAuth* parameter to
8702 determine what action shall be taken next.

8703 The ESI shall resend the *PublishCPPEvent* command, but with the *CPPAuth* field now set to the
8704 value received in the *CPPEventResponse* command.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

8705

8706 D.4.2.3.16 **GetCreditPayment Command**

8707 This command initiates *PublishCreditPayment* commands for the requested credit payment
8708 information.

8709 D.4.2.3.16.1 Payload Format

8710 The *GetCreditPayment* command payload shall be formatted as illustrated in Figure D-73.

Octets	**4**	**1**
Data Type	UTCTime	Unsigned 8-bit Integer
Field Name	Latest End Time (M)	NumberOf Records (M)

8711 **Figure D-73– *GetCreditPayment* Command Payload**

8712 D.4.2.3.16.2 Payload Details

8713 **Latest End Time (mandatory)**: UTC timestamp indicating the latest *CreditPaymentDate*
8714 of records to be returned by the corresponding *PublishCreditPayment* commands. The
8715 first returned *PublishCreditPayment* command shall be the most recent record with its
8716 *CreditPaymentDate* equal to or older than the *Latest End Time* provided.

8717 **NumberofRecords (mandatory)**: An 8-bit integer that represents the maximum number of
8718 *PublishCreditPayment* commands that the client is willing to receive in response to this
8719 command. A value of 0 would indicate all available *PublishCreditPayment* commands shall be
8720 returned. If more than one record is requested, the *PublishCreditPayment* commands should be
8721 returned with descending ordered *CreditPaymentDate*. If fewer records are available than are
8722 being requested, only those available are returned.

8723 D.4.2.3.16.3 Effect on Receipt

8724 A ZCL Default response with status NOT_FOUND shall be returned if there is no credit
8725 payment information available.

8726

8727 D.4.2.3.17 **GetCurrencyConversion Command**

8728 This command initiates a *PublishCurrencyConversion* command for the currency conversion
8729 factor updates. A server shall be capable of storing both the old and the new currencies.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

8730 D.4.2.3.17.1 Payload Details

8731 This command has no payload.

8732 D.4.2.3.17.2 Effect on Receipt

8733 A ZCL Default response with status NOT_FOUND shall be returned if there are no currency
8734 conversion factor updates available.

8735

8736 D.4.2.3.18 **GetTariffCancellation Command**

8737 This command initiates the return of the last *CancelTariff* command held on the associated
8738 server.

8739 D.4.2.3.18.1 Payload Details

8740 This command has no payload.

8741 D.4.2.3.18.2 When Generated

8742 This command is generated when the client device wishes to fetch any pending *CancelTariff*
8743 command from the server (see D.4.2.4.15 for further details). In the case of a BOMD, this may
8744 be as a result of the associated Notification flag.

8745 A ZCL Default response with status NOT_FOUND shall be returned if there is no *CancelTariff*
8746 command available.

8747

8748 ## D.4.2.4 Commands Generated

8749 The server side of the Price cluster is capable of generating the commands listed in Table D-96.

8750 **Table D-96– Generated Command IDs for the Price Cluster**

Command Identifier Field Value	Description	Mandatory / Optional
0x00	*Publish Price*	M
0x01	*Publish Block Period*	O
0x02	*Publish Conversion Factor*	O
0x03	*Publish Calorific Value*	O
0x04	*PublishTariffInformation*	O
0x05	*PublishPriceMatrix*	O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x06	*PublishBlockThresholds*	O
0x07	*PublishCO_2Value*	O
0x08	*PublishTierLabels*	O
0x09	*PublishBillingPeriod*	O
0x0A	*PublishConsolidatedBill*	O
0x0B	*PublishCPPEvent*	O
0x0C	*PublishCreditPayment*	O
0x0D	*PublishCurrencyConversion*	O
0x0E	*CancelTariff*	O
0x0F – 0xFF	Reserved	

8751

8752 D.4.2.4.1 **Publish Price Command**

8753 The *Publish Price* command is generated in response to receiving a *Get Current Price*
8754 command (see sub-clause D.4.2.3.2), in response to a *Get Scheduled Prices* command (see sub-
8755 clause D.4.2.3.3), and when an update to the pricing information is available from the
8756 commodity provider, either before or when a TOU price becomes active. Additionally the Publish
8757 Price Command is generated as specified in sub-clause D.4.4.3 when Block Pricing is in effect.

8758 When a *Get Current Price* or *Get Scheduled Prices* command is received over a ZigBee Smart
8759 Energy network, the *Publish Price* command should be sent unicast to the requester. In the
8760 case of an update to the pricing information from the commodity provider, the *Publish Price*
8761 command should be unicast to all individually registered devices implementing the Price
8762 Cluster on the ZigBee Smart Energy network.

8763 Devices capable of receiving this command must be capable of storing and supporting at
8764 least two pricing information instances, the current active price and the next price. By supporting
8765 at least two pricing information instances, receiving devices will allow the *Publish Price*
8766 command generator to publish the next pricing information during the current pricing period.

8767 Nested and overlapping *Publish Price* commands are not allowed. The current active price
8768 will be replaced if new price information is received by the ESI. In the case of overlapping
8769 events, the event with the newer Issuer Event ID takes priority over all nested and overlapping
8770 events. All existing events that overlap, even partially, should be removed. The only exception to
8771 this is that if an event with a newer Issuer Event ID overlaps with the end of the current active
8772 price but is not yet active, the active price is not deleted but its duration is modified to
8773 0xFFFF (until changed) so that the active price ends when the new event begins.

8774 D.4.2.4.1.1 Payload Format

8775 The *PublishPrice* command payload shall be formatted as illustrated in Figure D-74.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Octets	4	1-13	4	4	1	2	1
Data Type	Unsigned 32-bit Integer	Octet String	Unsigned 32-bit Integer	UTCTime	8 bits enumeration	Unsigned 16-bit Integer	8-bit BitMap
Field Name	Provider ID (M)	Rate Label (M)	Issuer Event ID (M)	Current Time (M)	Unit of Measure (M)	Currency (M)	Price Trailing Digit & Price Tier (M)

8776

Octets	1	4	2	4	1	4	1
Data Type	8-bit BitMap	UTCTime	Unsigned 16-bit Integer	Unsigned 32-bit Integer	Unsigned 8-bit Integer	Unsigned 32-bit Integer	Unsigned 8-bit Integer
Field Name	Number of Price Tiers & Register Tier (M)	Start Time (M)	Duration In Minutes (M)	Price (M)	Price Ratio (O)	Generation Price (O)	Generation Price Ratio (O)

8777

Octets	4	1	1	1	1
Data Type	Unsigned 32-bit Integer	8-bit Enumeration	8-bit BitMap	8 bit Integer	8-bit BitMap
Field Name	Alternate Cost Delivered (O)	Alternate Cost Unit (O)	Alternate Cost Trailing Digit(O)	Number of Block Thresholds (O)	Price Control (O)

8778

Octets	1	1	1	1	1
Data Type	8 bit Integer	8-bit Enumeration	8-bit Enumeration	8-bit Enumeration	8-bit Enumeration
Field Name	Number of Generation Tiers(O)	Generation Tier(O)	Extended Number of Price Tiers (O)	Extended Price Tier (O)	Extended Register Tier (O)

8779 **Figure D-74– *Publish Price* Command Payload**

8780 ***Note:*** *M = Mandatory field, O = Optional field.* ***All fields must be present in the payload.***
8781 *Optional fields will be marked with specific values to indicate they are not being used.*

8782 **Provider ID (mandatory):** An unsigned 32-bit field containing a unique identifier for
8783 the commodity provider. This field allows differentiation in deregulated markets where multiple
8784 commodity providers may be available.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Rate Label (mandatory): A ZCL Octet String field capable of storing a 12 character string (the first Octet indicates length) containing commodity provider- specific information regarding the current billing rate. The String shall be encoded in the UTF-8 format. This field allows differentiation when a commodity provider may have multiple pricing plans.

Issuer Event ID (mandatory): Unique identifier generated by the commodity provider. When new pricing information is provided that replaces older pricing information for the same time period, this field allows devices to determine which information is newer. It is expected that the value contained in this field is a unique number managed by upstream servers or a UTC based time stamp (UTCTime data type) identifying when the *Publish Price* command was issued. Thus, newer pricing information will have a value in the Issuer Event ID field that is larger than older pricing information.

Current Time (mandatory): A UTCTime field containing the current time as determined by the device. This field provides an extra value-added feature for the broadcast price signals.

Unit of Measure (mandatory): An 8-bit enumeration field identifying the commodity as well as its base unit of measure. The enumeration used for this field shall match one of the UnitOfMeasure values using a pure binary format as defined in the Metering cluster (see sub-clause D.3.2.2.4.1).

Currency (mandatory): An unsigned 16-bit field containing identifying information concerning the local unit of currency used in the price field. This field allows the displaying of the appropriate symbol for a currency (i.e.: $).

The value of the currency field should match the values defined by ISO 4217.

Price Trailing Digit and Price Tier (mandatory): An 8-bit field used to determine where the decimal point is located in the price field and to indicate the current pricing tier as chosen by the commodity provider. The most significant nibble is the Trailing Digit sub-field which indicates the number of digits to the right of the decimal point. The least significant nibble is an enumerated field containing the current Price Tier.

Valid values for the Price Tier sub-field are from 1 to 15 reflecting the least expensive tier (1) to the most expensive tiers (15). A value of zero indicates no price tier is in use. This parameter also references the associated *TiernPriceLabel* attribute assigned to the Price Tier. Table D-97 depicts the assignments. The meaning of value 0xF is dependant on the value of the optional *Extended Price Tier* field. Absence of this field, or a value of 0x00 in this field, indicates that the current Price Tier is fifteen, and references the *Tier15PriceLabel* attribute.. Where the *Extended Price Tier* field contains a non-zero value, the current Price Tier and *TiernPriceLabel* attribute are determined by the sum of the values of the *Price Tier* sub-field and the *Extended Price Tier* field.

Table D-97– Price Tier Sub-field Enumerations

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Enumerated Value	Price Tier
0x0	No Tier Related
0x1	Reference *Tier1PriceLabel*
0x2	Reference *Tier2PriceLabel*
0x3	Reference *Tier3PriceLabel*
0x4	Reference *Tier4PriceLabel*
0x5	Reference *Tier5PriceLabel*
0x6	Reference *Tier6PriceLabel*
0x7	Reference *Tier7PriceLabel*
0x8	Reference *Tier8PriceLabel*
0x9	Reference *Tier9PriceLabel*
0xA	Reference *Tier10PriceLabel*
0xB	Reference *Tier11PriceLabel*
0xC	Reference *Tier12PriceLabel*
0xD	Reference *Tier13PriceLabel*
0xE	Reference *Tier14PriceLabel*
0xF	Dependant on the value of the *Extended Price Tier* field

8820

8821 **Number of Price Tiers & Register Tier (mandatory):** An 8-bit BitMap where the most
8822 significant nibble is an enumerated sub-field representing the maximum number of price tiers
8823 available, and the least significant nibble is an enumerated sub-field indicating the register tier
8824 used with the current Price Tier.

8825 Valid values for the Number of Price Tiers sub-field are from 0 to 15 reflecting no tiers in use (0)
8826 to fifteen or more tiers available (15). The meaning of value 0xF is dependant on the value of the
8827 optional the *Extended Number of Price Tiers* field. Absence of this field, or a value of 0x00 in
8828 this field, indicates that maximum number of tiers available is fifteen. Where the *Extended*
8829 *Number of Price Tiers* field contains a non-zero value, the maximum number of tiers available is
8830 determined by the sum of the values of the *Number of Price Tiers* sub-field and the *Extended*
8831 *Number of Price Tiers* field.

8832 The Register Tier values correlate which *CurrentTierNSummationDelivered* attribute, found in
8833 sub-clause D.3.2.2.2, is accumulating usage informationRegister Tier enumerated values are
8834 listed in Table D-98. The meaning of value 0xF is dependant on the value of the optional
8835 *Extended Register Tier* field. Absence of this field, or a value of 0x00 in this field, indicates that
8836 usage information is being accumulated in the *CurrentTier15SummationDelivered* attribute.
8837 Where the *Extended Register Tier* field contains a non-zero value, the

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

CurrentTierNSummationDelivered attribute currently accumulating usage information by the sum of the values of the *Register Tier* sub-field and the *Extended Register Tier* field.

Both attributes can be used to calculate and display usage and subsequent costs.

Table D-98– Register Tier Sub-field Enumerations

Enumerated Value	Register Tier
0x0	No Tier Related
0x1	Usage accumulating in *CurrentTier1SummationDelivered* attribute
0x2	Usage accumulating in *CurrentTier2SummationDelivered* attribute
0x3	Usage accumulating in *CurrentTier3SummationDelivered* attribute
0x4	Usage accumulating in *CurrentTier4SummationDelivered* attribute
0x5	Usage accumulating in *CurrentTier5SummationDelivered* attribute
0x6	Usage accumulating in *CurrentTier6SummationDelivered* attribute
0x7	Usage accumulating in *CurrentTier7SummationDelivered* attribute
0x8	Usage accumulating in *CurrentTier8SummationDelivered* attribute
0x9	Usage accumulating in *CurrentTier9SummationDelivered* attribute
0xA	Usage accumulating in *CurrentTier10SummationDelivered* attribute
0xB	Usage accumulating in *CurrentTier11SummationDelivered* attribute
0xC	Usage accumulating in *CurrentTier12SummationDelivered* attribute
0xD	Usage accumulating in *CurrentTier13SummationDelivered* attribute
0xE	Usage accumulating in *CurrentTier14SummationDelivered* attribute
0xF	Dependant on the value of the *Extended Register Tier* field

Start Time (mandatory): A UTCTime field to denote the time at which the price signal becomes valid. A Start Time of 0x00000000 is a special time denoting "now."

If the device would send a price with a Start Time of now, adjust the Duration In Minutes field to correspond to the remainder of the price.

Duration In Minutes (mandatory): An unsigned 16-bit field used to denote the amount of time in minutes after the Start Time during which the price signal is valid. Maximum value means "until changed". If Block Charging only is in use (see sub-clause D.4.4.3 for further details), the Duration in Minutes field of the Publish Price command shall be set to 0xFFFF indicating the price is valid "until changed".

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Price (mandatory): An unsigned 32-bit field containing the price of the commodity measured in base unit of Currency per Unit of Measure with the decimal point located as indicated by the Price Trailing Digit field when the commodity is delivered to the premises.

Price Ratio (optional): An unsigned 8-bit field that gives the ratio of the price denoted in the Price field to the "normal" price chosen by the commodity provider. This field is thought to be useful in situations where client devices may simply be interested in pricing levels or ratios. The value in this field should be scaled by a factor of 0.1, giving a range of ratios from 0.1 to 25.4. A value of 0xFF indicates the field is not used and 0x00 is an invalid value.

Generation Price (optional): An unsigned 32-bit field containing the price of the commodity measured in base unit of Currency per Unit of Measure with the decimal point located as indicated by the Price Trailing Digit field when the commodity is received from the premises. An example use of this field is in energy markets where the price of electricity from the grid is different than the price of electricity placed on the grid. A value of 0xFFFFFFFF indicates the field is not used.

Generation Price Ratio (optional): An unsigned 8-bit field that gives the ratio of the price denoted in the Generation Price field to the "normal" price chosen by the commodity provider. This field is thought to be useful in situations where client devices may simply be interested in pricing levels or ratios. The value in this field should be scaled by a factor of 0.1, giving a range of ratios from 0.1 to 25.4 A value of 0xFF indicates the field is not used and 0x00 is an invalid value.

Alternate Cost Delivered (optional): An unsigned 32-bit Integer field that provides a mechanism to describe an alternative measure of the cost of the energy consumed. An example of an Alternate Cost might be the emissions of CO_2 for each kWh of electricity consumed providing a measure of the environmental cost. Another example is the emissions of CO_2 for each cubic meter of gas consumed (for gas metering). A different value for each price tier may be provided which can be used to reflect the different mix of generation that is associated with different TOU rates. A value of 0xFFFFFFFF indicates the field is not used.

Alternate Cost Unit (optional): An 8-bit enumeration identifying the unit (as specified in Table D-99) for the Alternate Cost Delivered field. A value of 0xFF indicates the field is not used.

Table D-99– Alternate Cost Unit Enumerations

Values	Description
0x00	Reserved for future use
0x01	Kg of CO_2 per unit of measure
0x02 to 0xFF	Reserved for future use

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Alternate Cost Trailing Digit (optional): An 8-bit BitMap field used to determine where the decimal point is located in the alternate cost field. The most significant nibble indicates the number of digits to the right of the decimal point. The least significant nibble is reserved. A value of 0xFF indicates the field is not used.

Number of Block Thresholds (optional): An 8-bit integer which indicates the number of block thresholds available. Valid values are from 0 to 15 reflecting no blocks in use (0) to 15 block thresholds available (15). A value of 0xFF indicates field not used. Any value between 1 and 15 indicates that Block Pricing shall be used, see sub-clause D.4.4.3 for further details.

For combined Block/TOU charging, where multiple sets of Block Thresholds are being utilized, the field shall indicate the number of block thresholds available in the current price tier.

Price Control (optional): Identifies additional control options for the price event. A value of 0x00 indicates field not used. Note that for ZigBee SE 1.1 and later devices, the *Price Acknowledgement* command is mandatory, but for SE 1.0 devices, it was optional, so the sender of the *Publish Price* command should not rely on receiving a *Price Acknowledgment* command even if the Price Acknowledgement bit in the Price Control Field is set.

If Bit 1 is set, this indicates that the total number of tiers exceeds the 15 specified in the command; this shall indicate to a client complying with this specification that it should read the total number of tiers using the *GetTariffInformation* command.

The BitMap for this field is described in Table D-100.

Table D-100– Price Control Field BitMap

Bit	Description
0	0=Price Acknowledgement not required, 1=Price Acknowledgement required
1	0=Total Tiers DOES NOT exceed 15 1= Total Tiers exceeds the 15 specified in the command
2 to 7	Reserved

Number of Generation Tiers (optional): Specifies the total number of generation tiers applicable in the current tariff, valid values are 0-48.

Generation Tier (optional): An 8-bit enumerated value specifying the current generation tier. See Table D-101.

Table D-101– Generation Tier Enumerations

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Enumerated Value	Description
0x00	Reserved
0x01	Usage accumulating in *CurrentTier1SummationReceived* attribute
0x02	Usage accumulating in *CurrentTier2SummationReceived* attribute
0x03	Usage accumulating in *CurrentTier3SummationReceived* attribute
0x04	Usage accumulating in *CurrentTier4SummationReceived* attribute
0x05	Usage accumulating in *CurrentTier5SummationReceived* attribute
0x06	Usage accumulating in *CurrentTier6SummationReceived* attribute
0x07	Usage accumulating in *CurrentTier7SummationReceived* attribute
0x08	Usage accumulating in *CurrentTier8SummationReceived* attribute
0x09	Usage accumulating in *CurrentTier9SummationReceived* attribute
0x0A	Usage accumulating in *CurrentTier10SummationReceived* attribute
0x0B	Usage accumulating in *CurrentTier11SummationReceived* attribute
0x0C	Usage accumulating in *CurrentTier12SummationReceived* attribute
0x0D	Usage accumulating in *CurrentTier13SummationReceived* attribute
0x0E	Usage accumulating in *CurrentTier14SummationReceived* attribute
0x0F	Usage accumulating in *CurrentTier15SummationReceived* attribute
0x10	Usage accumulating in *CurrentTier16SummationReceived* attribute
0x11	Usage accumulating in *CurrentTier17SummationReceived* attribute
0x12	Usage accumulating in *CurrentTier18SummationReceived* attribute
0x13	Usage accumulating in *CurrentTier19SummationReceived* attribute
0x14	Usage accumulating in *CurrentTier20SummationReceived* attribute
0x15	Usage accumulating in *CurrentTier21SummationReceived* attribute
0x16	Usage accumulating in *CurrentTier22SummationReceived* attribute
0x17	Usage accumulating in *CurrentTier23SummationReceived* attribute
0x18	Usage accumulating in *CurrentTier24SummationReceived* attribute
0x19	Usage accumulating in *CurrentTier25SummationReceived* attribute
0x1A	Usage accumulating in *CurrentTier26SummationReceived* attribute
0x1B	Usage accumulating in *CurrentTier27SummationReceived* attribute
0x1C	Usage accumulating in *CurrentTier28SummationReceived* attribute
0x1D	Usage accumulating in *CurrentTier29SummationReceived* attribute
0x1E	Usage accumulating in *CurrentTier30SummationReceived* attribute
0x1F	Usage accumulating in *CurrentTier31SummationReceived* attribute
0x20	Usage accumulating in *CurrentTier32SummationReceived* attribute
0x21	Usage accumulating in *CurrentTier33SummationReceived* attribute
0x22	Usage accumulating in *CurrentTier34SummationReceived* attribute
0x23	Usage accumulating in *CurrentTier35SummationReceived* attribute
0x24	Usage accumulating in *CurrentTier36SummationReceived* attribute
0x25	Usage accumulating in *CurrentTier37SummationReceived*attribute
0x26	Usage accumulating in *CurrentTier38SummationReceived*attribute

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

8910

8911

0x27	Usage accumulating in *CurrentTier39SummationReceived* attribute
0x28	Usage accumulating in *CurrentTier40SummationReceived* attribute
0x29	Usage accumulating in *CurrentTier41SummationReceived*attribute
0x2A	Usage accumulating in *CurrentTier42SummationReceived* attribute
0x2B	Usage accumulating in *CurrentTier43SummationReceived* attribute
0x2C	Usage accumulating in *CurrentTier44SummationReceived* attribute
0x2D	Usage accumulating in *CurrentTier45SummationReceived* attribute
0x2E	Usage accumulating in *CurrentTier46SummationReceived* attribute
0x2F	Usage accumulating in *CurrentTier47SummationReceived* attribute
0x30	Usage accumulating in *CurrentTier48SummationReceived* attribute
0x31 – 0xFF	Reserved

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Extended Number of Price Tiers (optional): Where the the maximum number of price tiers available exceeds the value of 15 supported by the *Number of Price Tiers* sub-field, this enumerated field is used in conjunction with the *Number of Price Tiers* sub-field to indicate the maximum number of price tiers available. Valid values for the *Extended Number of Price Tiers* field are from 1 to 33, indicating a maximum number of tiers available from 16 to 48 respectively. A value of zero indicates that the maximum number of price tiers available is indicated by the *Number of Price Tiers* sub-field alone.

Extended Price Tier (optional): Where the current Price Tier exceeds the value of 15 supported by the *Price Tier* sub-field, this enumerated field is used in conjunction with the *Price Tier* sub-field to indicate the current Price Tier. Valid values for the *Extended Price Tier* field are from 1 to 33, indicating a current Price Tier of 16 to 48 respectively as shown in Table D-102. A value of zero indicates that the current status of the Price Tier is indicated by the *Price Tier* sub-field alone.

Table D-102– Extended Price Tier Field Enumerations

Enumerated Value	Price Tier
0x00	Refer to *Price Tier* sub-field
0x01	Reference *Tier16PriceLabel*
0x02	Reference *Tier17PriceLabel*
0x03	Reference *Tier18PriceLabel*
0x04	Reference *Tier19PriceLabel*
0x05	Reference *Tier20PriceLabel*
0x06	Reference *Tier21PriceLabel*
0x07	Reference *Tier22PriceLabel*
0x08	Reference *Tier23PriceLabel*
0x09	Reference *Tier24PriceLabel*
0x0A	Reference *Tier25PriceLabel*
0x0B	Reference *Tier26PriceLabel*
0x0C	Reference *Tier27PriceLabel*
0x0D	Reference *Tier28PriceLabel*
0x0E	Reference *Tier29PriceLabel*
0x0F	Reference *Tier30PriceLabel*
0x10	Reference *Tier31PriceLabel*
0x11	Reference *Tier32PriceLabel*
0x12	Reference *Tier33PriceLabel*
0x13	Reference *Tier34PriceLabel*
0x14	Reference *Tier35PriceLabel*
0x15	Reference *Tier36PriceLabel*
0x16	Reference *Tier37PriceLabel*
0x17	Reference *Tier38PriceLabel*

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x18	Reference *Tier39PriceLabel*
0x19	Reference *Tier40PriceLabel*
0x1A	Reference *Tier41PriceLabel*
0x1B	Reference *Tier42PriceLabel*
0x1C	Reference *Tier43PriceLabel*
0x1D	Reference *Tier44PriceLabel*
0x1E	Reference *Tier45PriceLabel*
0x1F	Reference *Tier46PriceLabel*
0x20	Reference *Tier47PriceLabel*
0x21	Reference *Tier48PriceLabel*

Extended Register Tier (mandatory): Where the current Register Tier exceeds the value of 15 supported by the *Register Tier* sub-field, this enumerated field is used in conjunction with the *Register Tier* sub-field to indicate which *CurrentTierNSummationDelivered* attribute, found in sub-clause D.3.2.2.2, is accumulating usage information. Valid values for the *Extended Register Tier* field are from 1 to 33, indicating a current Register Tier of 16 to 48 respectively as shown in Table D-103. A value of zero indicates that the current status of the Register Tier is indicated by the *Register Tier* sub-field alone.

Table D-103– Extended Register Tier Field Enumerations

Enumerated Value	Register Tier
0x00	Refer to *Register Tier* sub-field
0x01	Usage accumulating in *CurrentTier16SummationDelivered* attribute
0x02	Usage accumulating in *CurrentTier17SummationDelivered* attribute
0x03	Usage accumulating in *CurrentTier18SummationDelivered* attribute
0x04	Usage accumulating in *CurrentTier19SummationDelivered* attribute
0x05	Usage accumulating in *CurrentTier20SummationDelivered* attribute
0x06	Usage accumulating in *CurrentTier21SummationDelivered* attribute
0x07	Usage accumulating in *CurrentTier22SummationDelivered* attribute
0x08	Usage accumulating in *CurrentTier23SummationDelivered* attribute
0x09	Usage accumulating in *CurrentTier24SummationDelivered* attribute
0x0A	Usage accumulating in *CurrentTier25SummationDelivered* attribute
0x0B	Usage accumulating in *CurrentTier26SummationDelivered* attribute
0x0C	Usage accumulating in *CurrentTier27SummationDelivered* attribute
0x0D	Usage accumulating in *CurrentTier28SummationDelivered* attribute
0x0E	Usage accumulating in *CurrentTier29SummationDelivered* attribute

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x0F	Usage accumulating in *CurrentTier30SummationDelivered* attribute
0x10	Usage accumulating in *CurrentTier31SummationDelivered* attribute
0x11	Usage accumulating in *CurrentTier32SummationDelivered* attribute
0x12	Usage accumulating in *CurrentTier33SummationDelivered* attribute
0x13	Usage accumulating in *CurrentTier34SummationDelivered* attribute
0x14	Usage accumulating in *CurrentTier35SummationDelivered* attribute
0x15	Usage accumulating in *CurrentTier36SummationDelivered* attribute
0x16	Usage accumulating in *CurrentTier37SummationDelivered* attribute
0x17	Usage accumulating in *CurrentTier38SummationDelivered* attribute
0x18	Usage accumulating in *CurrentTier39SummationDelivered* attribute
0x19	Usage accumulating in *CurrentTier40SummationDelivered* attribute
0x1A	Usage accumulating in *CurrentTier41SummationDelivered* attribute
0x1B	Usage accumulating in *CurrentTier42SummationDelivered* attribute
0x1C	Usage accumulating in *CurrentTier43SummationDelivered* attribute
0x1D	Usage accumulating in *CurrentTier44SummationDelivered* attribute
0x1E	Usage accumulating in *CurrentTier45SummationDelivered* attribute
0x1F	Usage accumulating in *CurrentTier46SummationDelivered* attribute
0x20	Usage accumulating in *CurrentTier47SummationDelivered* attribute
0x21	Usage accumulating in *CurrentTier48SummationDelivered* attribute

8935

8936 D.4.2.4.1.2 Effect on Receipt

8937 On receipt of this command, the device is informed of a price event for the specific
8938 provider, commodity, and currency indicated.

8939 Should the device choose to change behavior based on the price event, the change of behavior
8940 should occur after a random delay between 0 and 5 minutes, to avoid potential spikes that could
8941 occur as a result of coordinated behavior changes. Likewise, should a device choose to
8942 change behavior based on the expiration of the price event, the change in behavior should occur
8943 after a random delay between 0 and 5 minutes.

8944

8945 D.4.2.4.2 **Publish Block Period Command**

8946 The *Publish Block Period* command is generated in response to receiving a *Get Block*
8947 *Period(s)* command (see sub-clause D.4.2.3.5) or when an update to the block tariff schedule is
8948 available from the commodity provider. When the *Get Block Period(s)* command is received
8949 over the ZigBee Smart Energy network, the *Publish Block Period* command(s) should be sent

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

unicast to the requestor. In the case of an update to the block tariff schedule from the commodity provider, the *Publish Block Period* command should be unicast to all individually registered devices implementing the Price Cluster on the ZigBee Smart Energy network.

Devices capable of receiving this command must be capable of storing and supporting two block periods, the current active block and the next block. By supporting two block periods, receiving devices will allow the *Publish Block Period* command generator to publish the next block information during the current block period.

D.4.2.4.2.1 Payload Format

Octets	**4**	**4**	**4**	**3**	**1**	**1**	**1**	**1**
Data Type	Unsigned 32 bit Integer	Unsigned 32 bit Integer	UTCTime	Unsigned 24 bit Integer	8 bit BitMap	8 bit BitMap	8 bit BitMap	8 bit Enumeration
Field Name	Provider ID (M)	Issuer Event ID (M)	Block Period Start Time (M)	Block Period Duration (M)	Block Period Control (M)	Block Period Duration Type (M)	Tariff Type (M)	Tariff Resolution Period (M)

Figure D-75– *Publish Block Period* Command Payload

Provider ID (mandatory): An unsigned 32-bit field containing a unique identifier for the commodity provider. This field allows differentiation in deregulated markets where multiple commodity providers may be available.

Issuer Event ID (mandatory): Unique identifier generated by the commodity provider. When new block period information is provided that replaces older information for the same period, this field allows devices to determine which information is newer. It is expected that the value contained in this field is a unique number managed by upstream servers or a UTC based time stamp (UTCTime data type) identifying when the *Publish Block Period* command was issued. Thus, newer block period information will have a value in the Issuer Event ID field that is larger than older block information.

Block Period Start Time (mandatory): A UTCTime field to denote the time at which the block tariff period starts. A start time of 0x00000000 is a special time denoting "now". If the device would send an event with a Start Time of now, adjust the Duration In Minutes field to correspond to the remainder of the event. A start date/time of 0xFFFFFFFF shall cause an existing *PublishBlockPeriod* command with the same *Provider ID* and *Issuer Event ID* to be cancelled (note that, in markets where permanently active price information is required for billing purposes, it is recommended that a replacement/superseding *Publish Block Period* command is used in place of this cancellation mechanism).

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Where the *Duration Timebase* is set to a value other than *Minutes*, the *Duration Control* sub-field provides further clarification; where *Duration Control* is set to *Start of Timebase*, the *Block Period Start Time* shall be set to 00:00:00 on the applicable date, and where *Duration Control* is set to *End of Timebase*, the *Block Period Start Time* shall be set to 23:59:59 on the applicable date.

Block Period Duration (mandatory): An unsigned 24-bit field to denote the block tariff period . The duration units are defined by the *Block Period Duration Type* field. Maximum value (0xFFFFFF) means 'until changed'.

Block Period Control (mandatory): Identifies additional control options for the block period event. A value of 0x00 indicates field not used.

The BitMap for this field is described in Table D-104.

Table D-104– Block Period Control Field BitMap

Bit	Description
0	1=Price Acknowledgement required, 0=Price Acknowledgement not required
1	1=Repeating Block, 0=Non Repeating Block
2-7	Reserved

Price Acknowledgement: Indicates whether a *Price Acknowledgment* command shall be returned on receipt of this *Publish Block Period* command.

Repeating Block: Indicates whether a block period repeats on expiry. Note that the interaction between Block and Billing periods is out of scope of this specification.

Block Period Duration Type (mandatory): An 8-bit bitmap where the least significant nibble is an enumerated sub-field indicating the time base used for the duration, and the most significant nibble is an enumerated sub-field providing duration control.

Enumerated values for the Duration Timebase are shown in Table D-105:

Table D-105– Block Period DurationTimebase Enumeration

Value	Description
0x0	Minutes (default)
0x1	Days
0x2	Weeks
0x3	Months
0x4-0xF	Reserved

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Enumerated values for the Duration Control are shown in Table D-106:

Table D-106– Block Period Duration Control Enumeration

Value	Description
0x0	Start of Timebase
0x1	End of Timebase
0x2	Not Specified
0x3-0xF	Reserved

Where the *Duration Timebase* is set to a value other than *Minutes*, the *Duration Control* sub-field provides further clarification; *Start of Timebase* indicates that the duration shall run from the START of the respective day, week or month, whereas *End of Timebase* shall indicate that the duration runs from the END of the respective day, week or month. The *Duration Control* sub-field shall be set to *Not Specified* when a timebase of *Minutes* is in use.

Tariff Type (mandatory): An 8-bit bitmap identifying the type of tariff published in this command. The least significant nibble represents an enumeration of the tariff type as detailed in Table D-108 (Generation Meters shall use the 'Received' Tariff). The most significant nibble is reserved.

Tariff Resolution Period (mandatory): An 8 bit enumeration identifying the resolution period for the block tariff. See Table D-107:

Table D-107– Tariff Resolution Period Enumeration

Value	Description
0x00	Not Defined
0x01	Block Period
0x02	1 Day
0x03-0xFF	Reserved

The Tariff Resolution of *Block Period* means that the Block Tariff is applied based on calculations to the Block Thresholds defined in the command set without smoothing.

The Tariff resolution period of *1 Day* means that the application should apply "daily resolution", with recalculation of the thresholds through the Block Period to achieve the same result for the end of the Block Period but smoothing out the tariff application for the customer. This is described as follows:

Daily resolution of block tariffs is a method by which customers on a block tariff are charged on the basis of assigning the block thresholds on a day in proportion to the period through the block

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

period. For example, if the Block Period is 90 days and the day is number 45 in the period, then the thresholds which determine the cost to date on that day will be 50% of the thresholds defined for the whole Block period. This creates an averaging effect on the block tariff and prevents the customer from being exposed to one or more potentially large cost changes for many days during the billing period which can create customer concern, particularly in prepayment applications, and replacing these with cost changes during each day which are less apparent, but create the same total charges.

D.4.2.4.3 **PublishConversionFactor Command**

The *PublishConversionFactor* command is sent in response to a *GetConversionFactor* command or if a new conversion factor is available.

Clients shall be capable of storing at least two instances of the Conversion Factor, the currently active one and the next one.

D.4.2.4.3.1 Payload Format

Octets	**4**	**4**	**4**	**1**
Data Type	Unsigned 32-bit Integer	UTC Time	Unsigned 32-bit Integer	8-bit BitMap
Field Name	Issuer Event ID (M)	Start Time (M)	Conversion Factor (M)	Conversion Factor Trailing Digit (M)

Figure D-76– *PublishConversionFactor* Command Payload

D.4.2.4.3.2 Payload Details

Issuer Event ID (mandatory): Unique identifier generated by the commodity provider.

Start Time (mandatory): A UTCTime field to denote the time at which the value becomes valid. The value remains valid until replaced by a newer one.

Conversion Factor (mandatory): See Price Cluster Commodity attributes (see sub-clause D.4.2.2.4.3).

Conversion Factor Trailing Digit (mandatory): See Price Cluster Commodity attributes (see sub-clause D.4.2.2.4.4).

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

9048 D.4.2.4.4 **PublishCalorificValue Command**

9049 The *PublishCalorificValue* command is sent in response to a *GetCalorificValue* command or if
9050 a new calorific value is available. Clients shall be capable of storing at least two instances of
9051 the Calorific Value, the currently active one and the next one.

9052 D.4.2.4.4.1 Payload Format

Octets	**4**	**4**	**4**	**1**	**1**
Data Type	Unsigned 32-bit Integer	UTC Time	Unsigned 32-bit Integer	8-bit Enumeration	8-bit BitMap
Field Name	Issuer Event ID (M)	Start Time (M)	Calorific Value (M)	Calorific Value Unit (M)	Calorific Value Trailing Digit (M)

9053 **Figure D-77– *PublishCalorificValue* Command Payload**

9054 D.4.2.4.4.2 Payload Details

9055 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider.

9056 **Start Time (mandatory):** A UTCTime field to denote the time at which the value becomes valid.
9057 The value remains valid until replaced by a newer one.

9058 **Calorific Value (mandatory):** See Price Cluster Commodity attributes (see sub- clause
9059 D.4.2.2.4.5).

9060 **Calorific Value Unit (mandatory):** See Price Cluster Commodity attributes (see sub-clause
9061 D.4.2.2.4.6).

9062 **Calorific Value Trailing Digit (mandatory):** See Price Cluster Commodity attributes (see
9063 sub-clause D.4.2.2.4.7).

9064

9065 D.4.2.4.5 **PublishTariffInformation Command**

9066 The *PublishTariffInformation* command is sent in response to a *GetTariffInformation* command
9067 or if new tariff information is available (including Price Matrix and Block Thresholds).

9068 Clients should be capable of storing at least **two** instances of the Tariff Information, the currently
9069 active and the next one. Note that there may be separate tariff information for consumption
9070 delivered and received.

9071 Note that the payload for this command could be up to 61 bytes in length, therefore
9072 fragmentation may be required.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

D.4.2.4.5.1 Payload Format

Octets	4	4	4	4	1	1..25	1	1
Data Type	Unsigned 32-bit Integer	Unsigned 32-bit Integer	Unsigned 32-bit Integer	UTC Time	8-bit BitMap	Octet String	Unsigned 8-bit Integer	Unsigned 8-bit Integer
Field Name	Provider ID (M)	Issuer Event ID (M)	Issuer Tariff ID (M)	Start Time (M)	Tariff Type / Charging Scheme (M)	Tariff Label (M)	Number of Price Tiers in Use(M)	Number of Block Thresholds in Use(M)

1	2	1	4	1	3	3
8-bit Enumeration	Unsigned 16-bit Integer	8-bit BitMap	Unsigned 32-bit Integer	Unsigned 8-bit Integer	Unsigned 24-bit Integer	Unsigned 24-bit Integer
Unit of Measure (M)	Currency (M)	Price Trailing Digit (M)	Standing Charge (M)	TierBlockMode (M)	Block Threshold Multiplier (M)	Block Threshold Divisor (M)

Figure D-78– *PublishTariffInformation* Command Payload

D.4.2.4.5.2 Payload Details

ProviderID (mandatory): A unique identifier for the commodity supplier. The *ProviderID* in this command will always be the one stored as the attribute (see D.10.2.2.1.1 or D.10.2.2.1.9 depending on *TariffType*) except for the case where a change of supplier is pending and the new supplier wishes to publish its tariff information in advance.

Issuer Event ID (mandatory): Unique identifier generated by the commodity provider. When new information is provided that replaces older information for the same time period, this field allows devices to determine which information is newer. The value contained in this field is a unique number managed by upstream servers or a UTC based time stamp (UTCTime data type) identifying when the Publish command was issued. Thus, newer information will have a value in the Issuer Event ID field that is larger than older information.

Issuer Tariff ID (mandatory): Unique identifier generated by the commodity supplier.

Start Time (mandatory): A UTC Time field to denote the time at which the price signal becomes valid. A start date/time of 0x00000000 shall indicate that the command should be executed immediately.

Tariff Type/Charging Scheme (mandatory): An 8-bit bitmap identifying the type of tariff published in this command. The least significant nibble represents an enumeration of the tariff type as detailed in Table D-108 (Generation Meters shall use the 'Received' Tariff), the most significant nibble represents an enumeration specifying the charging scheme as detailed in Table D-109.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

9096 **Table D-108– Tariff Type Enumeration**

Value	Description
0x0	Delivered Tariff
0x1	Received Tariff
0x2	Delivered and Received Tariff
0x3-0xF	Reserved

9097

9098 **Table D-109– Tariff Charging Scheme Enumeration**

Value	Description
0x0	TOU Tariff
0x1	Block Tariff
0x2	Block/TOU Tariff with common thresholds
0x3	Block/TOU Tariff with individual thresholds per tier
0x4-0xF	Reserved.

9099

9100 **Tariff Label (mandatory):** The format and use of this field is the same as for the *TariffLabel*
9101 attribute or *ReceivedTariffLabel* attribute (depending on *TariffType*) as defined inD.4.2.2.7.1
9102 and D.4.2.2.15.1 respectively.

9103 **Number of Price Tiers in Use (mandatory):** The format and use of this field is the same as for
9104 the *NumberofPriceTiersInUse* attribute or *ReceivedNumberofPriceTiersInUse* attribute
9105 (depending on *TariffType/Charging Scheme*) as defined in D.4.2.2.7.2 and D.4.2.2.15.2
9106 respectively.

9107 **Number of Block Thresholds in Use (mandatory):** The format and use of this field is the same
9108 as for the *NumberofBlockThresholdsInUse* attribute or *ReceivedNumberofBlockThresholdsInUse*
9109 attribute (depending on *TariffType/Charging Scheme*) as defined in D.4.2.2.7.3 and D.4.2.2.15.3
9110 respectively.

9111 **Unit of Measure (mandatory):** The format and use of this field is the same as for the *Unit of*
9112 *Measure* attribute as defined in D.4.2.2.7.5.

9113 **Currency (mandatory):** The format and use of this field is the same as for the *Currency*
9114 attribute as defined in D.4.2.2.7.6.

9115 **Price Trailing Digit (mandatory):** The format and use of this field is the same as for the
9116 *PriceTrailingDigit* attribute as defined in D.4.2.2.7.7.

9117 **Standing Charge (mandatory):** The format and use of this field is the same as for the
9118 *StandingCharge* attribute as defined in D.4.2.2.4.2. A value of 0xFFFFFFFF indicates the field is

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

not used. When publishing Received tariffs (according to *TariffType*) this field should be set to 0xFFFFFFFF.

TierBlockMode (mandatory): The format and use of this field is the same as for the *TierBlockMode* attribute or *ReceivedTierBlockMode* attribute (depending on *TariffType*) as defined in D.4.2.2.7.4 and D.4.2.2.15.4 respectively. In case of TOU or Block Charging only, this field is not used and shall be set to 0xFF. For combined Block/TOU charging, this field is mandatory and must be set to a valid value.

BlockThresholdMultiplier (mandatory): *BlockThresholdMultiplier* provides a value to be multiplied against Threshold parameter(s). If present, this attribute must be applied to all Block Threshold values to derive values that can be compared against the *CurrentBlockPeriodConsumptionDelivered* attribute within the Metering cluster. This parameter must be used in conjunction with the *BlockThresholdDivisor* parameter(s). In case no multiplier is defined, this field shall be set to 1.

BlockThresholdDivisor (mandatory): *BlockThresholdDivisor* provides a value to divide the result of applying the *ThresholdMultiplier* attribute to Block Threshold values to derive values that can be compared against the *CurrentBlockPeriodConsumptionDelivered* attribute within the Metering cluster. This attribute must be used in conjunction with the *BlockThresholdMultiplier* parameter(s). In case no divisor is defined, this field shall be set to 1.

D.4.2.4.5.3 Effect on Receipt

If the client is unable to store this *PublishTariffInformation* command, the device should respond using a ZCL Default Response with a status of INSUFFICIENT_SPACE.

D.4.2.4.6 PublishPriceMatrix Command

The *PublishPriceMatrix* command is used to publish the Block Price Information Set (up to 15 tiers x 15 blocks) and the Extended Price Information Set (up to 48 tiers). The *PublishPriceMatrix* command is sent in response to a *GetPriceMatrix* command.

Clients should be capable of storing at least **two** instances of the Price Matrix, the currently active and the next one.

There may be a separate Price Matrix for consumption delivered and received; in this case, each Price Matrix will be identified by a different *IssuerTariffId* value.

The Price server shall send only the number of tiers and blocks as defined in the corresponding *PublishTariffInformation* command (*NumberofPriceTiersinUse*, *NumberofBlockThresholdsinUse*+1).

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

The maximum application payload may not be sufficient to transfer all Price Matrix elements in one command. Therefore the ESI may send as many *PublishPriceMatrix* commands as needed. In this case the first command shall have *CommandIndex* set to 0, the second to 1 and so on; all associated commands shall use the same value of *Issuer Event ID*. Note that, in this case, it is the client's responsibility to ensure that it receives all associated *PublishPriceMatrix* commands before any of the payloads can be used.

D.4.2.4.6.1 Payload Format

The *PublishPriceMatrix* command shall be formatted as illustrated in Figure D-79:

Octets	**4**	**4**	**4**	**4**	**1**	**1**	**1**	**Variable**
Data Type	Unsigned 32-bit Integer	Unsigned 32-bit Integer	UTC Time	Unsigned 32-bit Integer	Unsigned 8-bit Integer	Unsigned 8-bit Integer	8-bit BitMap	
Field Name	Provider ID (M)	Issuer Event ID (M)	Start Time (M)	Issuer Tariff ID (M)	Command Index (M)	Total Number of Commands (M)	Sub-payload Control	Price Matrix Sub-payload

Figure D-79– *PublishPriceMatrix* Command Payload

D.4.2.4.6.2 Payload Details

Provider ID (mandatory): An unsigned 32-bit field containing a unique identifier for the commodity provider. This field allows differentiation in deregulated markets where multiple commodity providers may be available.

Issuer Event ID (mandatory): Unique identifier generated by the commodity provider. When new information is provided that replaces older information for the same time period, this field allows devices to determine which information is newer. The value contained in this field is a unique number managed by upstream servers or a UTC based time stamp (UTCTime data type) identifying when the Publish command was issued. Thus, newer information will have a value in the Issuer Event ID field that is larger than older information.

Start Time (mandatory): A UTC Time field to denote the time at which the price signal becomes valid. A start date/time of 0x00000000 shall indicate that the command should be executed immediately.

Issuer Tariff ID (mandatory): Unique identifier generated by the commodity supplier. This must match the *Issuer Tariff ID* sent in the related *PublishTariffInformation* command.

Command Index (mandatory): The *Command Index* is used to count the payload fragments in the case that an entire payload does not fit into one message. The *Command Index* starts at 0 and is incremented for each fragment belonging to the same command.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Total Number of Commands (mandatory): In the case that an entire payload does not fit into one message, the *Total Number of Commands* field indicates the total number of sub-commands in the message.

Sub-Payload Control (mandatory): An 8-bit bitmap, the least significant bit of which specifies the information type stored in the sub payload (see Table D-110). The remaining bits are reserved.

Table D-110– PublishPriceMatrix Sub-Payload Control Bitmap

Bit	Description
0	0 = The information stored in the sub payload is Block only or Block/TOU based 1 = The information stored in the sub payload is TOU based.
1-7	Reserved

D.4.2.4.6.2.1 PriceMatrix Sub-Payload

Octets	1	4	1	4	…
Data Type	Unsigned 8-bit Integer	Unsigned 32- bit Integer	Unsigned 8-bit Integer	Unsigned 32-bit Integer	…
Field Name	Tier/Block ID (n)	Price(n)	Tier/Block ID (n+1)	Price(n+1)	…

Figure D-80– *PriceMatrix* Command Sub-Payload

Tier/Block ID (Mandatory): The Tier/Block ID specifies the TOU Tier or the TOU Tier and Block that the subsequent Price field in the command applies to.If Bit 0 of the *Sub-Payload Control* field is set to Zero, then the least significant nibble represents a value specifying the block number and the most significant nibble represents the Tier that the subsequent *Price* field applies to. Valid values for the Block Number sub-field are 0 to 15 reflecting block 1 (0) to block 16(15).Valid values for the Tiers sub-field are from 0 to 15 reflecting no tiers to tier fifteen.

If Bit 0 of the *Sub-Payload Control* field is set to one, then the field is an 8-bit value specifying the TOU Tier that the subsequent *Price* field applies to. Valid values are 1 to 48.

Price (Mandatory): This field holds the price information for the Block/TOU or TOU identified by the previous *Tier/Block ID* field. The price information is provided in a base unit of *Currency* with the decimal point located as indicated by the *Trailing Digits* field of a *PublishTariffInformation* command or by the attribute defined in the *Tariff Information Attribute Set*.

NOTE: *The number of blocks in use is one greater than the number of block thresholds in use. For TOU charging only (number of block thresholds in use = 0, number of blocks in use = 1), the price information of block 1, tier 1 to 15 shall be used.*

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

9206 D.4.2.4.7 **PublishBlockThresholds Command**

9207 The *PublishBlockThresholds* command is sent in response to a *GetBlockThresholds* command.

9208 Clients should be capable of storing at least **two** instances of the Block Thresholds, the currently
9209 active and the next one.

9210 There may be a separate set of Block Thresholds for consumption delivered and received; in this
9211 case, each set of Block Thresholds will be identified by a different *IssuerTariffId* value.

9212 The price server shall send only the number of block thresholds in use
9213 (*NumberofBlockThresholdsInUse*) as defined in the *PublishTariffInformation* command.

9214 The maximum application payload may not be sufficient to transfer all thresholds in one
9215 command. In this case the Price server may send two consecutive *PublishBlockThreshold*
9216 commands (*CommandIndex* set to 0 and 1 respectively); both commands shall use the same value
9217 of *Issuer Event ID*. Note that, in this case, it is the client's responsibility to ensure that it receives
9218 all associated *PublishBlockThreshold* commands before any of the payloads can be used.

9219 D.4.2.4.7.1 Payload Format

9220 The *PublishBlockThresholds* command shall be formatted as illustrated in Figure D-81:

Octets	**4**	**4**	**4**	**4**	**1**	**1**	**1**	**Variable**
Data Type	Unsigned 32-bit Integer	Unsigned 32-bit Integer	UTC Time	Unsigned 32-bit Integer	Unsigned 8-bit Integer	Unsigned 8-bit Integer	8-bit BitMap	Variable
Field Name	Provider ID (M)	Issuer Event ID (M)	Start Time (M)	Issuer Tariff ID (M)	Command Index (M)	Total Number of Commands (M)	Sub-payload Control (M)	Block Threshold Sub-payload

9221 **Figure D-81– *PublishBlockThresholds* Command Payload**

9222 D.4.2.4.7.2 Payload Details

9223 **Provider ID (mandatory):** An unsigned 32-bit field containing a unique identifier for
9224 the commodity provider. This field allows differentiation in deregulated markets where multiple
9225 commodity providers may be available.

9226 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. When
9227 new information is provided that replaces older information for the same time period, this field
9228 allows devices to determine which information is newer. The value contained in this field is a
9229 unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)
9230 identifying when the Publish command was issued. Thus, newer information will have a value in
9231 the *Issuer Event ID* field that is larger than older information.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Start Time (mandatory): A UTC Time field to denote the time at which the price signal becomes valid. A start date/time of 0x00000000 shall indicate that the command should be executed immediately.

Issuer Tariff ID (mandatory): Unique identifier generated by the commodity supplier. This must match the *Issuer Tariff ID* sent in the related *PublishTariffInformation* command.

Command Index (mandatory): The *Command Index* is used to count the payload fragments in the case where the entire payload does not fit into one message. The *Command Index* starts at 0 and is incremented for each fragment belonging to the same command.

Total Number of Commands (mandatory): In the case where the entire payload does not fit into one message, the *Total Number of Commands* field indicates the total number of sub-commands in the message.

Sub-Payload Control (Mandatory): The *Sub-Payload Control* bitmap specifies the usage of the information contained within the *Block Threshold Sub-Payload* (see Table D-111).

Table D-111– PublishBlockThresholds Sub-Payload Control Bitmap

Bit	Description
0	0 = Block Thresholds supplied apply to a specific TOU tier. 1 = Block Thresholds supplied apply to all TOU tiers or when Block Only charging is in operation
1 - 7	Reserved

D.4.2.4.7.2.1 BlockThreshold Sub-Payload

The *BlockThreshold Sub-Payload* consists of multiple sets of data which consist of a Tier ID, Block Threshold Count and the threshold values associated with the stated Tier. The number of thresholds contained in any one set is identified in the *NumberOfBlockThresholds* sub-field.

Octets	**1**	**6**	**…**	**6**	**1**	**6**	**…**	**6**	**…**
Data Type	8-bit Bitmap	Unsigned 48-bit Integer	…	Unsigned 48-bit Integer	8-bit Bitmap	Unsigned 48-bit Integer	…	Unsigned 48-bit Integer	…
Field Name	Tier / NumberOf Block Thresholds (M)	Block Threshold 1 (M)	…	Block Threshold n (M)	Tier / NumberOf Block Thresholds (M)	Block Threshold 1	…	Block Threshold n	…

Figure D-82– BlockThreshold Sub-Payload Format

Tier/NumberOfBlockThresholds: The *Tier/NumberOfBlockThresholds* field is an 8 bitmap. The format of the bitmap is decided by bit0 of the sub-payload control field.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

If Bit0 of the *Sub-Payload Control* field is 0, then the least significant nibble represents a value specifying the number of thresholds to follow in the command. The most significant nibble represents the Tier that the subsequent block threshold values apply to.

If Bit0 of the *Sub-Payload Control* field is 1, then the most significant nibble is unused and should be set to 0.

Valid values for the *NumberOfBlockThresholds* sub-field are 0 to 15 reflecting no block in use (0) to block 15(15).Valid values for the Tiers sub-field are from 0 to 15 reflecting no tier to tier fifteen.

If the thresholds for a particular tier (Bit0 of the *Sub-Payload Control* field is 0) or the total number of thresholds (Bit0 of the *Sub-Payload Control* field is 1) will not fit into a single *PublishBlockThresholds* command, then the value of this *NumberOfBlockThresholds* sub-field shall indicate the number of thresholds of the relevant type contained within this particular command only.

BlockThreshold: The Block Thresholds represent the threshold values applicable to an individual block period and, where applicable, to a particular tier.

The thresholds are established such that crossing the threshold of energy consumption for the present block activates the next higher block, which can affect the energy rate in a positive or negative manner. The values are absolute and always increasing. The values represent the threshold at the end of a block. The Unit of Measure will be based on the fields defined in the *PublishTariffInformation* command, the formatting being defined by *ThresholdDivisor* and *ThresholdMultiplier*.

D.4.2.4.8 **PublishCO_2Value Command**

The *PublishCO_2Value* command is sent in response to a *GetCO_2Value* command or if a new CO_2 conversion factor is available.

Clients should be capable of storing at least **two** instances of the CO_2 conversion factor, the currently active and the next one.

D.4.2.4.8.1 Payload Format

The *PublishCO_2Value* command shall be formatted as illustrated in Figure D-83:

Octets	**4**	**4**	**4**	**1**	**4**	**1**	**1**
Data Type	Unsigned 32-bit Integer	Unsigned 32-bit Integer	UTC Time	8-bit Bitmap	Unsigned 32-bit Integer	8- bit Enumeration	8-bit BitMap

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Field Name	Provider ID (M)	Issuer Event ID (M)	Start Time (M)	Tariff Type (M)	CO_2 Value (M)	CO_2 Value Unit (M)	CO_2 Value Trailing Digit (M)

Figure D-83– *PublishCO$_2$Value* Command Payload

D.4.2.4.8.2 Payload Details

Provider ID (mandatory): An unsigned 32-bit field containing a unique identifier for the commodity provider. This field allows differentiation in deregulated markets where multiple commodity providers may be available.

Issuer Event ID (mandatory): Unique identifier generated by the commodity provider. When new information is provided that replaces older information for the same time period, this field allows devices to determine which information is newer. The value contained in this field is a unique number managed by upstream servers or a UTC based time stamp (UTCTime data type) identifying when the Publish command was issued. Thus, newer information will have a value in the *Issuer Event ID* field that is larger than older information.

Start Time (mandatory): A UTC Time field to denote the time at which the CO_2 value becomes valid. A start date/time of 0x00000000 shall indicate that the command should be executed immediately. A start date/time of 0xFFFFFFFF shall cause an existing *PublishCO$_2$Value* command with the same *Provider ID* and *Issuer Event ID* to be cancelled (note that, in markets where permanently active price information is required for billing purposes, it is recommended that a replacement/superseding *PublishCO$_2$Value* command is used in place of this cancellation mechanism).

Tariff Type (mandatory): An 8-bit bitmap identifying the type of tariff published in this command. The least significant nibble represents an enumeration of the tariff type as detailed in Table D-108 (Generation Meters shall use the 'Received' Tariff). The most significant nibble is reserved.

CO_2 Value (mandatory): The format and use of this field is the same as for the *CO_2* attribute or *ReceivedCO$_2$* attribute (depending on *TariffType*) as defined in D.4.2.2.7.9 and D.4.2.2.15.6 respectively. A value of 0xFFFFFFFF indicates field not used.

CO_2 Unit (mandatory): The format and use of this field is the same as for the *CO$_2$Unit* attribute or *ReceivedCO$_2$Unit* attribute (depending on *TariffType*) as defined in D.4.2.2.7.10 and D.4.2.2.15.7 respectively. A value of 0xFF indicates field not used.

CO_2 Trailing Digit (mandatory): The format and use of this field is the same as for the *CO$_2$TrailingDigit* attribute or *ReceivedCO$_2$TrailingDigit* attribute (depending on *TariffType*) as defined in D.4.2.2.7.11 and D.4.2.2.15.8 respectively. A value of 0xFF indicates field not used.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

9314 D.4.2.4.9 **PublishTierLabels Command**

9315 The *PublishTierLabels* command is generated in response to receiving a *GetTierLabels*
9316 command or when there is a tier label change.

9317 D.4.2.4.9.1 Payload Format

Octets	**4**	**4**	**4**	**1**	**1**	**1**	**1**	**1-13**
Data Type	Unsigned 32-bit Integer	Unsigned 32-bit Integer	Unsigned 32-bit Integer	Unsigned 8-bit Integer	Unsigned 8-bit Integer	Unsigned 8-bit Integer	Unsigned 8-bit Integer	Octet string
Field Name	Provider ID (M)	Issuer Event ID (M)	Issuer Tariff ID (M)	Command Index (M)	Total Number of Commands (M)	Number of Labels(M)	Tier ID	TierLabel

9318

Octets	**…**	**1**	**1-13**
Data Type	…	Unsigned 8-bit Integer	Octet string
Field Name	…	Tier ID (number of labels -1)	TierLabel (number of labels -1)

9319 **Figure D-84– *PublishTierLabels* Command Payload**

9320 D.4.2.4.9.2 Payload Details

9321 **Provider ID (mandatory):** An unsigned 32-bit field containing a unique identifier for
9322 the commodity provider. This field allows differentiation in deregulated markets where multiple
9323 commodity providers may be available.

9324 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. When
9325 new information is provided that replaces older information for the same time period, this field
9326 allows devices to determine which information is newer. It is expected that the value contained in
9327 this field is a unique number managed by upstream servers or a UTC based time stamp
9328 (UTCTime data type) identifying when the Publish command was issued. Thus, newer
9329 information will have a value in the *Issuer Event ID* field that is larger than older information.

9330 **Issuer Tariff ID (mandatory):** Unique identifier generated by the commodity supplier. This is
9331 used to identify the tariff that the labels apply to.

9332 **Command Index (mandatory):** The *Command Index* is used to count the payload fragments in
9333 the case where the entire payload does not fit into one message. The *Command Index* starts at 0
9334 and is incremented for each fragment belonging to the same command.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Total Number of Commands (mandatory): In the case where the entire payload does not fit into one message, the *Total Number of Commands* field indicates the total number of sub-commands in the message.

Number of Labels (mandatory): The number of *Tier ID*/*Tier Label* sets contained within the command.

Tier ID (mandatory): The tier number that the associated *Tier Label* applies to.

Tier Label (mandatory): ZCL Octet String field capable of storing a 12 character string (the first character indicates the string length, represented in hexadecimal format) encoded in the UTF-8 format.

D.4.2.4.10 PublishBillingPeriod Command

The *PublishBillingPeriod* command is generated in response to receiving a *GetBillingPeriod*(s) command or when an update to the Billing schedule is available from the commodity supplier.

Nested and overlapping *PublishBillingPeriod* commands are not allowed. In the case of overlapping billing periods, the period with the newer *IssuerEventID* takes priority over all nested and overlapping periods. All existing periods that overlap, even partially, should be removed. Note however that there may be separate billing schedules for consumption delivered and received.

D.4.2.4.10.1 Payload Format

Octets	**4**	**4**	**4**	**3**	**1**	**1**
Data Type	Unsigned 32-bit Integer	Unsigned 32-bit Integer	UTC Time	Unsigned 24-bit Integer	8-bit Bitmap	8-bit Bitmap
Field Name	Provider ID (M)	Issuer Event ID (M)	Billing Period Start Time (M)	Billing Period Duration (M)	Billing Period Duration Type (M)	Tariff Type (M)

Figure D-85– *PublishBillingPeriod* Command Payload

D.4.2.4.10.2 Payload Details

Provider ID (mandatory): An unsigned 32-bit field containing a unique identifier for the commodity provider. This field allows differentiation in deregulated markets where multiple commodity providers may be available.

Issuer Event ID (mandatory): Unique identifier generated by the commodity provider. When new information is provided that replaces older information for the same time period, this field allows devices to determine which information is newer. The value contained in this field is a unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

identifying when the Publish command was issued. Thus, newer information will have a value in the *Issuer Event ID* field that is larger than older information.

Billing Period Start Time (mandatory): A UTC Time field to denote the time at which the billing period starts. A start time of 0x00000000 is a special time denoting "now". A start date/time of 0xFFFFFFFF shall cause an existing *PublishBillingPeriod* command with the same *Provider ID* and *Issuer Event ID* to be cancelled (note that, in markets where permanently active price information is required for billing purposes, it is recommended that a replacement/superseding *PublishBillingPeriod* command is used in place of this cancellation mechanism).

Billing Period Duration (mandatory): An unsigned 24-bit field to denote the billing period duration. The duration units are defined by the *Billing Period Duration Type* field.

Billing periods are always repeating, i.e. after *BillingPeriodDuration* has elapsed since a *BillingPeriodStartTime*, a new billing period will start with the same duration.

Billing Period Duration Type (mandatory): An 8-bit bitmap where the least significant nibble is an enumerated sub-field indicating the time base used for the duration, and the most significant nibble is an enumerated sub-field providing duration control. Enumerated values for the Duration Timebase are shown in Table D-105. Enumerated values for the Duration Control are shown in Table D-106.

Tariff Type (mandatory): An 8-bit bitmap identifying the type of tariff published in this command. The least significant nibble represents an enumeration of the tariff type as detailed in Table D-108 (Generation Meters shall use the 'Received' Tariff). The most significant nibble is reserved.

D.4.2.4.11 **PublishConsolidatedBill Command**

The *PublishConsolidatedBill* command is used to make consolidated billing information from previous billing periods available to other end devices. This command is issued in response to a *GetConsolidatedBill* command or if new billing information is available.

Nested and overlapping *PublishConsolidatedBill* commands are not allowed. In the case of overlapping consolidated bills, the bill with the newer *IssuerEventID* takes priority over all nested and overlapping bills. All existing bills that overlap, even partially, should be removed. Note however that there may be separate consolidated bills for consumption delivered and received.

A server device shall be capable of storing **five** consolidated bill command events as a minimum.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

9396 D.4.2.4.11.1 Payload Format

Octets	4	4	4	3	1	1	4	2	1
Data Type	Unsigned 32-bit Integer	Unsigned 32-bit Integer	UTC Time	Unsigned 24-bit Integer	8-bit Bitmap	8-bit Bitmap	Unsigned 32-bit Integer	Unsigned 16-bit Integer	8 bit BitMap
Field Name	Provider ID (M)	Issuer Event ID (M)	Billing Period Start Time (M)	Billing Period Duration (M)	Billing Period Duration Type (M)	Tariff Type (M)	Consolidated Bill (M)	Currency (M)	Bill Trailing Digit (M)

9397 **Figure D-86– *PublishConsolidatedBill* Command Payload**

9398 D.4.2.4.11.2 Payload Details

9399 **Provider ID (mandatory):** An unsigned 32-bit field containing a unique identifier for
9400 the commodity provider. This field allows differentiation in deregulated markets where multiple
9401 commodity providers may be available.

9402 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. When
9403 new information is provided that replaces older information for the same time period, this field
9404 allows devices to determine which information is newer. The value contained in this field is a
9405 unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)
9406 identifying when the Publish command was issued. Thus, newer information will have a value in
9407 the *Issuer Event ID* field that is larger than older information.

9408 **Billing Period Start Time (mandatory):** A UTC Time field containing the start time of the
9409 related billing period. A start date/time of 0x00000000 shall indicate that the command should be
9410 executed immediately. A start date/time of 0xFFFFFFFF shall cause an existing
9411 *PublishConsolidatedBill* command with the same *Provider ID* and *Issuer Event ID* to be
9412 cancelled (note that, in markets where permanently active price information is required for
9413 billing purposes, it is recommended that a replacement/superseding *PublishConsolidatedBill*
9414 command is used in place of this cancellation mechanism).

9415 **Billing Period Duration (mandatory):** An unsigned 24-bit field denoting the duration of the
9416 related billing period. The duration units are defined by the *Billing Period Duration Type* field.

9417 **Billing Period Duration Type (mandatory):** An 8-bit bitmap where the least significant nibble
9418 is an enumerated sub-field indicating the time base used for the duration, and the most significant
9419 nibble is an enumerated sub-field providing duration control. Enumerated values for the Duration
9420 Timebase are shown in Table D-105. Enumerated values for the Duration Control are shown in
9421 Table D-106.

9422 **Tariff Type (mandatory):** An 8-bit bitmap identifying the type of tariff published in this
9423 command. The least significant nibble represents an enumeration of the tariff type as detailed in

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Table D-108 (Generation Meters shall use the 'Received' Tariff). The most significant nibble is reserved.

Consolidated Bill (mandatory): An unsigned 32-bit field containing the consolidated bill value for the stated billing period. The *Consolidated Bill* field should be provided in the same currency as used in the Price cluster.

Currency (mandatory): An unsigned 16-bit field containing identifying information concerning the local unit of currency used in the *Consolidated Bill* field.

The value of the currency field should match the values defined by ISO 4217.

BillTrailingDigit (mandatory): An 8-bit field used to determine where the decimal point is located in the *Consolidated Bill* field. The most significant nibble contains the *Trailing Digit* sub-field which indicates the number of digits to the right of the decimal point.

D.4.2.4.12 PublishCPPEvent Command

Note: *The PublishCPPEvent command in this revision of this specification is provisional and not certifiable. This feature may change before reaching certifiable status in a future revision of this specification.*

The *PublishCPPEvent* command is sent from an ESI to its Price clients to notify them of a Critical Peak Pricing (CPP) event.

D.4.2.4.12.1 Payload Format

Octets	**4**	**4**	**4**	**2**	**1**	**1**	**1**
Data Type	Unsigned 32-bit Integer	Unsigned 32-bit Integer	UTC Time	Unsigned 16-bit Integer	8-bit Bitmap	8-bit Enumeration	8-bit Enumeration
Field Name	Provider ID (M)	Issuer Event ID (M)	Start Time (M)	Duration in Minutes (M)	Tariff Type (M)	CPP Price Tier (M)	CPP Auth (M)

Figure D-87– *PublishCPPEvent* Command Payload

D.4.2.4.12.2 Payload Details

Provider ID (mandatory): An unsigned 32-bit field containing a unique identifier for the commodity provider. This field allows differentiation in deregulated markets where multiple commodity providers may be available.

Issuer Event ID (mandatory): Unique identifier generated by the commodity provider. When new information is provided that replaces older information for the same time period, this field

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

allows devices to determine which information is newer. The value contained in this field is a unique number managed by upstream servers or a UTC based time stamp (UTCTime data type) identifying when the Publish command was issued. Thus, newer information will have a value in the *Issuer Event ID* field that is larger than older information.

Start Time (mandatory): A UTC Time field to denote the time at which the CPP event begins. A start date/time of 0x00000000 shall indicate that the command should be executed immediately. A start date/time of 0xFFFFFFFF shall cause an existing *PublishCPPEvent* command with the same *Provider ID* and *Issuer Event ID* to be cancelled (note that, in markets where permanently active price information is required for billing purposes, it is recommended that a replacement/superseding *PublishCPPEvent* command is used in place of this cancellation mechanism).

Duration in Minutes: Defines the duration of the CPP event.

Tariff Type (mandatory): An 8-bit bitmap identifying the type of tariff published in this command. The least significant nibble represents an enumeration of the tariff type as detailed in Table D-108 (Generation Meters shall use the 'Received' Tariff). The most significant nibble is reserved.

CPP Price Tier (mandatory): An 8-bit enumeration identifying the price tier associated with this CPP event. The price(s) contained in the active price matrix for that price tier will override the normal pricing scheme. Prices 'CPP1' and 'CPP2' are reserved for this purposes (see D.4.2.2.6 for further details).

Table D-112– CPP Auth Enumeration

Value	Description
0	'CPP1'
1	'CPP2'
2 - 255	Reserved

CPP Auth (mandatory): An 8-bit enumeration identifying the status of the CPP event:

Table D-113– CPP Auth Enumeration

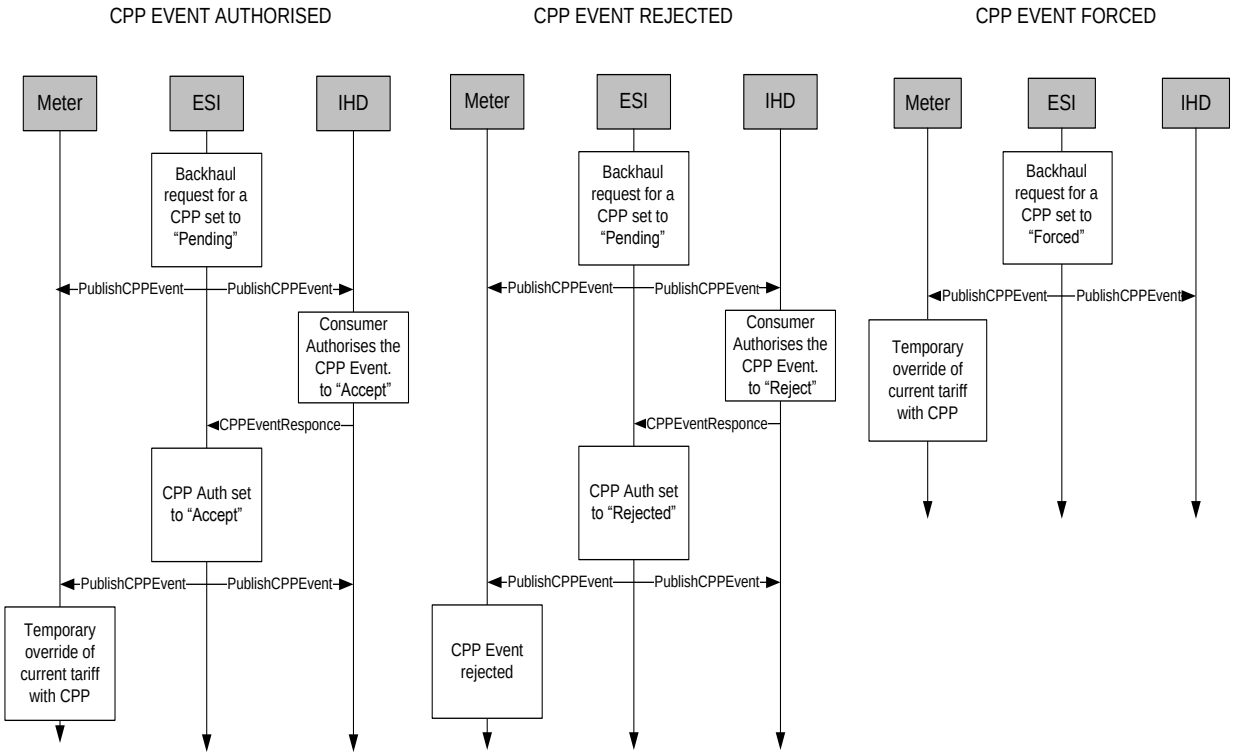
Value	Description
0	Pending
1	Accepted
2	Rejected
3	Forced
4 - 255	Reserved

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

9475 D.4.2.4.12.3 When Generated

9476 The *PublishCPPEvent* command is generated when the energy provider has requested the
9477 consumer to accept a CPP, when the consumer has accepted the CPP, or if the ESI has received a
9478 *CPPEventResponse* command. See Figure D-88.

9479



9480
9481 **Figure D-88– CPP Event Flow**
9482

9483 D.4.2.4.12.4 Effect on Receipt

9484 When the *PublishCPPEvent* command is received, the IHD or Meter shall act in one of two
9485 ways:

9486 1. It shall notify the consumer that there is a CPP event that requires acknowledgment. The
9487 acknowledgement shall be either to accept the *CPPEvent* or reject the *CPPEvent* (in which
9488 case it shall send the *CPPEventResponse* command, with the *CPPAuth* parameter set to
9489 Accepted or Rejected). It is recommended that the CPP event is ignored until a consumer
9490 either accepts or rejects the event.

9491 2. The *CPPAuth* parameter is set to "Forced", in which case the *CPPEvent* has been accepted.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

9492

9493 D.4.2.4.13 **PublishCreditPayment Command**

9494 The *PublishCreditPayment* command is used to update the credit payment information when
9495 available.

9496 Nested and overlapping *PublishCreditPayment* commands are not allowed. In the case of
9497 overlapping credit payments, the payment with the newer *Issuer Event ID* takes priority over all
9498 nested and overlapping payments. All existing payments that overlap, even partially, should be
9499 removed.

9500 A server device shall be capable of storing **five** credit payments command events as a minimum.

9501 D.4.2.4.13.1 Payload Format

Octets	**4**	**4**	**4**	**4**	**1**	**4**	**4**	**1-21**
Data Type	Unsigned 32-bit Integer	Unsigned 32-bit Integer	UTC Time	Unsigned 32-bit Integer	8-bit Enumeration	Unsigned 32-bit Integer	UTC Time	Octet String
Field Name	Provider ID (M)	Issuer Event ID (M)	Credit Payment Due Date (M)	Credit Payment Overdue Amount (M)	Credit Payment Status (M)	Credit Payment (M)	Credit Payment Date (M)	Credit Payment Ref (M)

9502 **Figure D-89– *PublishCreditPayment* Command Payload**

9503 D.4.2.4.13.2 Payload Details

9504 **Provider ID (mandatory):** An unsigned 32-bit field containing a unique identifier for
9505 the commodity provider. This field allows differentiation in deregulated markets where multiple
9506 commodity providers may be available.

9507 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. When
9508 new information is provided that replaces older information for the same time period, this field
9509 allows devices to determine which information is newer. The value contained in this field is a
9510 unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)
9511 identifying when the Publish command was issued. Thus, newer information will have a value in
9512 the *Issuer Event ID* field that is larger than older information.

9513 **Credit Payment Due Date (mandatory):** A UTC Time field containing the time that the next
9514 credit payment is due. See also section D.4.2.2.9.1.

9515 **Credit Payment Overdue Amount (mandatory):** An unsigned 32-bit field denoting the current
9516 amount this is overdue from the consumer. This field should be provided in the same currency as
9517 used in the Price cluster. See also section D.4.2.2.9.3.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Credit Payment Status (mandatory): An 8-bit enumeration identifying the current credit payment status. Refer to section D.4.2.2.9.2 for the format of this enumeration.

Credit Payment (mandatory): An unsigned 32-bit field denoting the last credit payment. This field should be provided in the same currency as used in the Price cluster. See also section D.4.2.2.9.6.

Credit Payment Date (mandatory): A UTC Time field containing the time at which the last credit payment was made. See also section D.4.2.2.9.7.

Credit Payment Ref (mandatory): An string of between 0-20 octets used to denote the last credit payment reference used by the energy supplier. See also section D.4.2.2.9.8.

D.4.2.4.14 **PublishCurrencyConversion Command**

The *PublishCurrencyConversion* command is sent in response to a *GetCurrencyConversion* command or when a new currency becomes available.

D.4.2.4.14.1 Payload Format

The *PublishCurrencyConversion* command shall be formatted as illustrated in the figure below:

Octets	**4**	**4**	**4**	**2**	**2**	**4**	**1**	**4**
Data Type	Unsigned 32-bit Integer	Unsigned 32-bit Integer	UTC Time	Unsigned 16-bit Integer	Unsigned 16-bit Integer	Unsigned 32-bit Integer	8-bit BitMap	32-bit BitMap
Field Name	Provider ID (M)	Issuer Event ID (M)	Start Time (M)	Old Currency (M)	New Currency (M)	Conversion Factor (M)	Conversion Factor Trailing Digit (M)	Currency Change Control Flags (M)

Figure D-90– *PublishCurrencyConversion* Command Payload

D.4.2.4.14.2 Payload Details

Provider ID (mandatory): An unsigned 32-bit field containing a unique identifier for the commodity provider. This field allows differentiation in deregulated markets where multiple commodity providers may be available.

Issuer Event ID (mandatory): Unique identifier generated by the commodity provider. When new information is provided that replaces older information for the same time period, this field allows devices to determine which information is newer. The value contained in this field is a unique number managed by upstream servers or a UTC based time stamp (UTCTime data type) identifying when the Publish command was issued. Thus, newer information will have a value in the *Issuer Event ID* field that is larger than older information.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Start Time (mandatory): A UTC Time field to denote the time at which the new currency becomes valid. A start date/time of 0x00000000 shall indicate that the command should be executed immediately. A start date/time of 0xFFFFFFFF shall cause an existing but pending *PublishCurrencyConversion* command with the same *Provider ID* and *Issuer Event ID* to be cancelled.

Old Currency (mandatory): An unsigned 16-bit field containing identifying information concerning the old local unit of currency used in the Price cluster. The value of the *Old Currency* field should match the values defined by ISO 4217.

New Currency (mandatory): An unsigned 16-bit field containing identifying information concerning the new local unit of currency used in the Price cluster. The value of the *New Currency* field should match the values defined by ISO 4217.

Conversion Factor (mandatory): The format and use of this field is the same as for the *ConversionFactor* attribute as defined in D.4.2.2.4.3.

Conversion Factor Trailing Digit (mandatory): The format and use of this field is the same as for the *ConversionFactorTrailingDigit* attribute as defined in D.4.2.2.4.4.

Currency Change Control Flags (mandatory): A 32-bit mask that denotes the functions that are required to be carried out on processing of this command. See Table D-114 below:

Table D-114– Currency Change Control

Bits	Description
0	1 = Clear Billing Information 0 = Do Not Clear Billing Information
1	1 = Convert Billing Information using the New Currency 0 = Do Not Convert Billing Information
2	1 = Clear Old Consumption Data 0 = Do Not Clear Old Consumption Data
3	1 = Convert Old Consumption Data using the New Currency 0 = Do Not Convert Old Consumption Data
4-31	Reserved

D.4.2.4.15 **CancelTariff Command**

The *CancelTariff* command indicates that all data associated with a particular tariff instance should be discarded.

In markets where permanently active price information is required for billing purposes, it is recommended that replacement/superseding *PublishTariffInformation*, *PublishPriceMatrix*,

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

PublishBlockThresholds and *PublishTierLabels* commands are used in place of a *CancelTariff* command.

D.4.2.4.15.1 Payload Format

The *CancelTariff* command shall be formatted as illustrated in Figure D-91:

Octets	**4**	**4**	**1**
Data Type	Unsigned 32-bit Integer	Unsigned 32-bit Integer	8-bit BitMap
Field Name	Provider ID (M)	Issuer Tariff ID (M)	Tariff Type (M)

Figure D-91– *CancelTariff* Command Payload

D.4.2.4.15.2 Payload Details

Provider ID (mandatory): An unsigned 32-bit field containing a unique identifier for the commodity provider. This field allows differentiation in deregulated markets where multiple commodity providers may be available.

Issuer Tariff ID (mandatory): Unique identifier generated by the commodity Supplier. All parts of a tariff instance shall have the same *Issuer Tariff ID*.

Tariff Type (mandatory): An 8-bit bitmap identifying the type of tariff to be cancelled by this command. The least significant nibble represents an enumeration of the tariff type as detailed in Table D-108 (Generation Meters shall use the 'Received' Tariff). The most significant nibble is reserved.

D.4.2.4.15.3 Effect on Receipt

On receipt of this command, a client device shall discard all instances of *PublishTariffInformation*, *PublishPriceMatrix*, *PublishBlockThresholds* and *PublishTierLabels* commands associated with the stated *Provider ID*, *Tariff Type* and *Issuer Tariff ID*.

D.4.3 Client

D.4.3.1 Dependencies

Events carried using this cluster include a timestamp with the assumption that target devices maintain a real time clock. Devices can acquire and synchronize their internal clocks via the ZCL Time server.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

If a device does not support a real time clock it is assumed that the device will interpret and utilize the "Start Now" 0x00000000 value within the Time field.

Note: *The Price Client Cluster Attributes in this revision of this specification are provisional and not certifiable. These features may change before reaching certifiable status in a future revision of this specification.*

D.4.3.2 Attributes

Table D-115– Price Client Cluster Attributes

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0000	*PriceIncrease Randomize Minutes*	Unsigned 8-bit Integer	0x00 to 0x3C	Read/ Write	0x05	O
0x0001	*PriceDecrease Randomize Minutes*	Unsigned 8-bit Integer	0x00 to 0x3C	Read/ Write	0x0F	O
0x0002	*Commodity Type*	8-bit Enumeration	0x00 to 0xFF	Read Only	-	O
0x0003 - 0x000F	Reserved					

D.4.3.2.1 **PriceIncreaseRandomizeMinutes Attribute**

The *PriceIncreaseRandomizeMinutes* attribute represents the maximum amount of time to be used when randomizing the response to a price increase. Note that although the granularity of the attribute is in minutes, it is recommended the granularity of the randomization used within a responding device be in seconds or smaller. If a device responds to a price increase it must choose a random amount of time, in seconds or smaller, between 0 and *PriceIncreaseRandomizeMinutes* minutes. The device must implement that random amount of time before or after the price change. How and if a device will respond to a price increase is up to the manufacturer. Whether to respond before or after the price increase is also up to the manufacturer.

As an example, a water heater with a *PriceIncreaseRandomizeMinutes* set to 6 could choose to lower its set point 315 seconds (but not more than 360 seconds) before the price increases.

The valid range for this attribute is 0x00 to 0x3C.

If *PriceIncreaseRandomizeMinutes* or *PriceDecreaseRandomizeMinutes* attributes are not supported by the client, then it should use the default values for the attributes as specified in the Price Client Cluster Attribute table.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

9615 D.4.3.2.2 **PriceDecreaseRandomizeMinutes Attribute**

9616 The *PriceDecreaseRandomizeMinutes* attribute represents the maximum number of minutes to
9617 be used when randomizing the response to a price decrease. Note that although the granularity
9618 of the attribute is in minutes, it is recommended the granularity of the randomization used within
9619 a responding device be in seconds or smaller. If a device responds to a price decrease it must
9620 choose a random amount of time, in seconds or smaller, between 0 and
9621 *PriceDecreaseRandomizeMinutes* minutes and implement that random amount of time before or
9622 after the price change. How and if a device will respond to a price decrease is up to the
9623 manufacturer. Whether to respond before or after the price increase is also up to the
9624 manufacturer.

9625 As an example, a dishwasher with a *PriceDecreaseRandomizeMinutes* set to 15 could choose
9626 to start its wash cycle 723 seconds (but not more than 900 seconds) after the price decreases.

9627 The valid range for this attribute is 0x00 to 0x3C.

9628 D.4.3.2.3 **CommodityType Attribute**

9629 *CommodityType* provides a label for identifying the type of pricing client present. The attribute is
9630 an enumerated value representing the commodity. The defined values are represented by the
9631 non-mirrored values (0-127) in the *MeteringDeviceType* attribute enumerations (refer to Table
9632 D-27).

9633 ### D.4.3.3 Commands Received

9634 The client receives the cluster-specific response commands detailed in sub-clause D.4.2.4.

9635 ### D.4.3.4 Commands Generated

9636 The client generates the cluster-specific commands detailed in sub-clause D.4.2.3, as required by
9637 the application.

9638

9639 ## D.4.4 Application Guidelines

9640 ### D.4.4.1 Registering for Commands

9641 Devices should use bind request to register for unsolicited *Publish Price*, *Display Message*
9642 and *Load Control Event* commands.

9643 ### D.4.4.2 Attribute Reporting

9644 Attribute reporting may be used for sending information in the Price Server Cluster
9645 Attributes table. The Price Cluster attributes can be polled periodically for updates. Polling
9646 should not occur more frequently than recommended in D.3.4.2. Use of the *Report Attribute*

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

command without report configuration may be used for unsolicited notification of an attribute value change. Sleepy devices may have to poll.

D.4.4.3 Block Tariffs

Upon reaching the *Start Time* of a received *Publish Price* command, a device's behavior will depend on the values of the *Number of Block Thresholds* and *Number of Price Tiers* fields. A client device needing to determine if it should use Block Pricing shall send a *Get Current Price* command to the Price server and check the *Number of Block Thresholds* in the Publish Price response. Any value between 1 and 15 indicates that Block Pricing shall be used.

The prices for a commodity being delivered to the premises shall be taken from the Block Pricing Information Attribute Set whenever Block Pricing is active.

D.4.4.3.1 **TOU Charging Only**

Indicated by the Number of Block Thresholds field being set to zero. Charging shall be according to the price fields within the *Publish Price* command itself.

D.4.4.3.2 **Block Charging only**

Indicated by the *Number of Price Tiers* fields being set to zero while the *Number of Block Thresholds* is between 0x01 and 0x0F.

A server shall not update the Block Threshold and Block Price attribute sets of an active Block Period. Updates to these attribute sets can only be done by creating a new Block Period. The server may create a new active Block Period by updating either *Block Period Start Time (*attribute *StartOfBlockPeriod*) alone or *Block Period Duration in Minutes* (attribute *BlockPeriodDuration*) followed by *Block Period Start Time (*attribute *StartOfBlockPeriod*) along with updating other attributes as desired.

When a server transmits a Publish Price command it shall additionally fill fields necessary to support backwards compatibility with clients that may not support Block Charging. The *Price* field shall be set according to the Block Price Information Attribute Set. The *Duration in Minutes* field shall be set to 0xFFFF indicating the price is valid “until changed”.

A server shall additionally transmit a Publish Price command to clients under the following conditions:

1. At the start of a Block Period
2. When it is notified that a Block Threshold has been crossed
3. When *Block Period Start Time* or *Block Period Duration in Minutes* have changed to indicate a new active block period

A client may cache attributes from the Block Threshold, Block Period, Block Price, and Billing Period attribute sets. Cached attributes are valid only during the active Block Period when received. Upon reaching *Block Period Start Time* or detecting a new active Block Period, the client should retrieve updated values for cached attributes.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

A client shall check for a new active Block Period on receipt of an asynchronous Publish Price command (i.e. not required on a Publish Price command in response to Get Current Price) by checking *Block Period Start Time* and *Block Period Duration in Minutes* for update. Additionally, it shall infrequently (e.g. once an hour) query the *StartOfBlockPeriod* and *BlockPeriodDuration* attributes to verify that the Block Period has not ended early.

D.4.4.3.3 Block/TOU Combination Charging

The *Number of Block Thresholds* and *Number of Price Tiers* fields will both be set to non-zero values, indicating the number of blocks and number of tiers respectively being used. The start of a Block period shall be indicated by the value of the *Block Period Start Time* field within a *Publish Block Period* command. If the currently active parameters are not already available on the client device then, upon reaching the *Block Period Start Time*, the attributes for the required number of Block Thresholds, together with the Block Prices for all required blocks for the selected tier should be fetched from the server. The *Block Period Duration in Minutes* field shall indicate the length of the block period.

A *Publish Price* command will be received for the start of each new TOU period during a block period. At this point the attributes for the Block Prices for all required blocks for the newly activated tier should be fetched from the server.

Devices shall cater for both 'blocks in tiers' and 'tiers in blocks' models. In either case, the relevant prices will be defined in the Block Pricing Information Attribute Set. The 'tiers in blocks' model will always implement a single set of block thresholds, whereas the 'blocks in tiers' model may implement different thresholds for each tier.

D.4.4.3.4 Application Guidelines for Block Pricing under specific events

HAN device not communicating with meter for extended period of time:

In this situation, when the HAN device reconnects with the meter, it will need to read the Block Information Set to calculate the correct cost for the given period. This is done by applying the prices for each block/tier combination to the consumption information for each block/tier combination. If a block period has passed while the HAN device was not communicating with the meter, then the prior period consumption information will not be known and the prior period cost cannot be calculated by the HAN device.

Meter installation or swap-out:

The new meter will need to be configured with the appropriate block thresholds, pricing, and block duration by the utility. If this does not occur precisely at the start of that customer's billing period, the utility will need to (a) pro-rate these amounts over the remaining billing period duration and (b) decide how to handle the initial portion of the period. Any information from the initial part of the billing period will be lost when the new meter is installed. As such,

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

HAN devices may not display accurate information for this billing period and utilities should advise customers of this situation. As a typical meter lifetime is expected to be in the range of 10 to 20 years, this event is expected to be rare.

D.4.4.4 Handling of Enhanced Tariffs

In 'Traditional' Smart Energy networks, the back-haul connection and Price server are incorporated into the meter. Fiscal accounting is out of scope of the ZigBee network. Indicative pricing information, determined by the utility or by an ESI using information supplied by the utility, is communicated from the Price server to other Smart Energy devices, using a *Publish Price* command, whenever the price changes.

In Smart Energy networks where the meter is detached from the back-haul connection, the meter often being battery-powered and therefore unable to communicate for the majority of the time, and specifically where a Prepayment meter is required to perform independent accounting functionality, there is a need for the meter to have local access to current price and price scheduling information at all times. The optional 'Enhanced' tariff mechanism described in this section provides functionality to satisfy this requirement.

An enhanced tariff consists of a number of commands. Depending on the mode of operation, an associated TOU calendar may also be required. *PublishBlockThresholds* and *PublishPriceMatrix* commands always include the number of block thresholds in use and number of blocks / tiers in use respectively. It is the responsibility of a client to fetch all parts belonging to a tariff after it has received an unsolicited *PublishTariffInformation* command. A client shall ensure that it successfully receives all commands associated with a tariff before any of the data for that tariff can be used. It is recommended that a client checks that the data received across all commands is valid.

Whenever a new tariff is made available to a Price Server, it shall send an unsolicited *PublishTariffInformation* command to its bound clients (BOMDs shall be notified via notification flags). Other parts of the tariff (*PriceMatrix* and *BlockThresholds*) are not sent unsolicited; the clients shall send corresponding *GetPriceMatrix* and *GetBlockThresholds* commands, as applicable, to fetch the required information from the server.

The Price Cluster supports different charging modes:

- TOU charging
- Block charging
- TOU/Block combination charging

D.4.4.4.1 **Block Charging**

In case of Block charging, the following information needs to be transferred from server to client:

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

1 PublishTariffInformation

2 PublishPriceMatrix (noTierBlock1 .. noTierBlockN)

3 PublishBlockThresholds

4 PublishBlockPeriod

In addition, a Gas-ESI may send a *PublishConversionFactor* or *PublishCalorificValue* command along with a tariff update, but does not necessarily need to.

D.4.4.4.2**TOU Charging**

In case of TOU charging, the following price information needs to be transferred from server to client:

1 PublishTariffInformation

2 PublishPriceMatrix (Tier1Block1 .. TierNBlock1)

3 PublishCalendar (see D.9 for further details)

In addition, a Gas-ESI may send a *PublishConversionFactor* or *PublishCalorificValue* command along with a tariff update, but does not necessarily need to.

Note: the TOU Calendar and the Tariff are linked by the start time and not by any IDs.

D.4.4.4.3**TOU/Block Charging**

In case of TOU/Block charging, the following price information needs to be transferred from server to client:

1 PublishTariffInformation

2 PublishPriceMatrix (Tier1Block1 .. TierNBlockM)

3 PublishBlockThresholds

4 PublishBlockPeriod

5 PublishCalendar (see D.9 for further details)

In addition, a Gas-ESI may send a *PublishConversionFactor* or *PublishCalorificValue* command along with a tariff update, but does not necessarily need to.

D.4.4.4.4 **Critical Peak Pricing**

Note: *The following application guidelines that pertain to Critical Peak Pricing in this revision of this specification are provisional and not certifiable. This text may change before reaching certifiable status in a future revision of this specification.*

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

The following additional guidelines hold for the usage of CPP events:

- The price tiers used for CPP events (via the *PublishCPPEvent* command) are treated in the price matrix just like the ones used in the TOU Calendar. In fact, nothing prevents a tariff scheme where the same price is employed at regular times through the TOU calendar and ad-hoc via CPP events. Two prices are reserved in the price matrix for CPP events, 'CPP1' and 'CPP2'

- ESIs conforming to these specifications need to send out a *Publish Price* command along with the *PublishCPPEvent* command, for Smart Energy devices that do not support the latter.

D.4.4.4.5 **Generation Charging**

All Generation meters shall use the 'Received' sections of the Price cluster to publish the tariff information, and the Received section of the Metering Cluster.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

D.5 Messaging Cluster

D.5.1 Overview

This cluster provides an interface for passing text messages between ZigBee devices. Messages are expected to be delivered via the ESI and then unicast to all individually registered devices implementing the Messaging Cluster on the ZigBee network, or just made available to all devices for later pickup. Nested and overlapping messages are not allowed. The current active message will be replaced if a new message is received by the ESI.

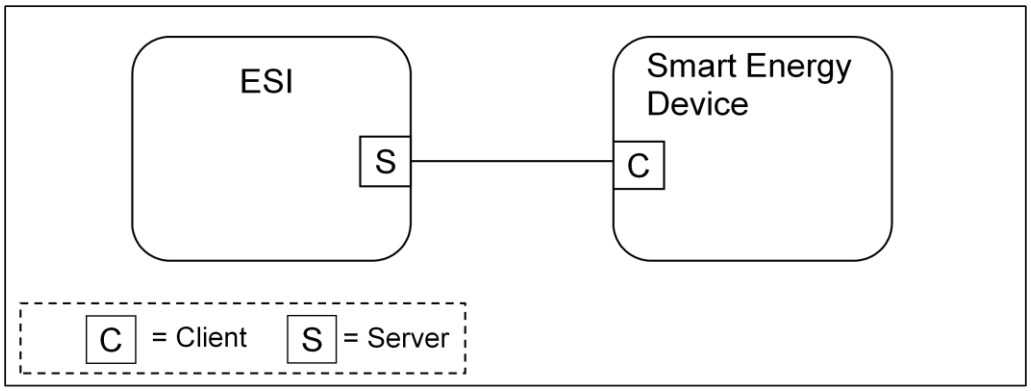


Figure D-92– Messaging Cluster Client/Server Example

Please note the ESI is defined as the Server due to its role in acting as the proxy for upstream message management systems and subsequent data stores.

D.5.2 Server

D.5.2.1 Dependencies

Support for ZCL Data Types.

No dependencies exist for other Smart Energy Clusters.

D.5.2.2 Attributes

None.

D.5.2.3 Commands Generated

The command IDs generated by the Messaging server cluster are listed in Table D-116.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

9817 **Table D-116– Generated Command IDs for the Messaging Server**

Command Identifier Field Value	Description	Mandatory / Optional
0x00	*Display Message*	M
0x01	*Cancel Message*	M
0x02	*Display Protected Message*	O
0x03	*Cancel All Messages*	O
0x04 – 0xff	Reserved	

9818

9819 D.5.2.3.1 **Display Message Command**

9820 D.5.2.3.1.1 Payload Format

9821 The *Display Message* command payload shall be formatted as illustrated in Figure D-93.

Octets	4	1	4	2	Variable	1
Data Type	Unsigned 32-bit integer	8-bit BitMap	UTCTime	Unsigned 16-bit Integer	Character string	8-bit BitMap
Field Name	Message ID (M)	Message Control (M)	Start Time (M)	Duration In Minutes (M)	Message (M)	Extended Message Control (O)

9822 **Figure D-93– *Display Message* Command Payload**

9823 ***D.5.2.3.1.1.1 Payload Details***

9824 **Message ID (mandatory)**: A unique unsigned 32-bit number identifier for this message. It's
9825 expected the value contained in this field is a unique number managed by upstream
9826 systems or a UTC based time stamp (UTCTime data type) identifying when the message was
9827 issued.

9828 **MessageControl (mandatory)**: An 8-bit BitMap field indicating control information related to the
9829 message. Bit encoding of this field is outlined in Table D-117:

9830 **Table D-117– Message Control Field Bit Map**

Bits	Enumeration	Value	Description
Bits 0 to 1	Normal transmission only	0	Send message through normal command function to client.
	Normal and Inter-PAN transmission	1	Send message through normal command function to client and pass message onto the Inter-PAN transmission mechanism.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

	Inter- PAN transmission only	2	Send message through the Inter- PAN transmission mechanism.
	Reserved	3	Reserved value for future use.
Bits 2 to 3	Low	0	Message to be transferred with a low level of importance.
	Medium	1	Message to be transferred with a medium level of importance.
	High	2	Message to be transferred with a high level of importance.
	Critical	3	Message to be transferred with a critical level of importance.
Bit 4	Reserved	N/A	This bit is reserved for future use.
Bit 5	Enhanced Confirmation Required	0	Enhanced Confirmation not required
		1	Enhanced Confirmation required
Bit 6	Reserved	N/A	This bit is reserved for future use.
Bit 7	Message Confirmation	0	Message Confirmation not required.
		1	Message Confirmation required.

Use of the Inter-PAN transmission mechanism within the Messaging cluster is now deprecated. A command where bits 0 to 1 indicate that it is for "Inter- PAN transmission only" shall be dropped; a ZCL Default Response command with a status of INVALID_FIELD shall be returned.

The *Message Confirmation* bit indicates the message originator requests a confirmation of receipt from a Utility Customer. If confirmation is required, the device should display the message or alert the user until it is either confirmed via a button, by selecting a confirmation option on the device, or the message expires. Confirmation is typically used when the Utility is sending down information such as a disconnection notice, or prepaid billing information.

The *Enhanced Confirmation Required* bit indicates that information is to be included in the confirmation of receipt from a Utility Customer ('YES', 'NO' or a text string). Earlier devices may treat bit 5 as reserved. In this case, these devices will assume that this bit is set to 0 (only basic confirmation required). Note that the *Message Confirmation* bit shall always be set whenever the *Enhanced Confirmation Required* bit is set.

Note: *It is desired that the device provide a visual indicator (flashing display or indicate with its LEDs as examples) that a message requiring confirmation is being displayed, and requires confirmation.*

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Start Time (mandatory): A UTCTime field to denote the time at which the message becomes valid. A Start Time of 0x00000000 is a special time denoting "now." If the device would send an event with a Start Time of now, adjust the Duration In Minutes field to correspond to the remainder of the event.

Duration In Minutes (mandatory): An unsigned 16-bit field is used to denote the amount of time in minutes after the Start Time during which the message is displayed. A Maximum value of 0xFFFF means "until changed".

Message (mandatory): A ZCL String containing the message to be delivered. The String shall be encoded in the UTF-8 format. Devices will have the ability to choose the methods for managing messages that are larger than can be displayed (truncation, scrolling, etc.).

For supporting larger messages sent over the network, both devices must agree upon a common Fragmentation ASDU Maximum Incoming Transfer Size. Please refer to sub-clause 5.3.8 for further details on Fragmentation settings.

Any message that needs truncation shall truncate on a UTF-8 character boundary. The SE secure payload is 59 bytes for the Message field in a non- fragmented, non-source routed Display Message packet (11 bytes for other Display Message fields). Devices using fragmentation can send a message larger than this. Reserving bytes for source route will reduce this.

ExtendedMessageControl (optional): An 8-bit BitMap field indicating additional control and status information for a given message. Bit encoding of this field is shown in Table D-118:

Table D-118– Extended Message Control Field Bit Map

Bit	Enumeration	Value	Description
Bit 0	Message Confirmation Status	0	Message has not been confirmed
		1	Message has been confirmed
Bits 1 - 7	Reserved for future use		

The *Message Confirmation Status* bit allows the confirmation state of a message to be communicated in the event that there are multiple IHD's (or other Messaging cluster client devices) on a network.

The server shall initially transmit a message requiring a confirmation with the *Message Confirmation Status* bit reset (0) to indicate the message had not yet been confirmed (the *Message Confirmation* bit of the *MessageControl* field will be set to indicate a confirmation is required).

When the message is confirmed on one of the multiple IHDs in the premises, a *Message Confirmation* command will be returned to the server. At this point, the server shall re-transmit

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

the original message, but with the *Message Confirmation Status* bit now set (1) to indicate that the message has been confirmed. This will indicate to other clients that the message no longer requires a confirmation.

D.5.2.3.2 Cancel Message Command

The *Cancel Message* command described in Figure D-94 provides the ability to cancel the sending or acceptance of previously sent messages. When this message is received the recipient device has the option of clearing any display or user interfaces it supports, or has the option of logging the message for future reference.

Octets	4	1
Data Type	Unsigned 32-bit integer	8-bit BitMap
Field Name	Message ID (M)	Message Control (M)

Figure D-94– *Cancel Message* Command Payload

D.5.2.3.2.1 Payload Details

Message ID (mandatory): A unique unsigned 32-bit number identifier for the message being cancelled. It's expected the value contained in this field is a unique number managed by upstream systems or a UTC based time stamp (UTCTime data type) identifying when the message was originally issued.

MessageControl (mandatory): This field is deprecated and should be set to 0x00.

D.5.2.3.3 Display Protected Message Command

The *Display Protected Message* command is for use with messages that are protected by a password or PIN.

D.5.2.3.3.1 Payload Format

The payload for this command shall be the same as that for a conventional *Display Message* command. See D.5.2.3.1.1.1 for payload details.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

9905 D.5.2.3.4 **Cancel All Messages Command**

9906 ***Note:*** *The Cancel All Messages command in this revision of this specification is provisional and*
9907 *not certifiable. This feature may change before reaching certifiable status in a future revision of*
9908 *this specification.*

9909 The *Cancel All Messages* command indicates to a client device that it should cancel all display
9910 messages currently held by it.

9911 D.5.2.3.4.1 Payload Format

Octets	**4**
Data Type	UTC Time
Field Name	Implementation Date/Time (M)

9912 **Figure D-95– *Cancel All Messages* Command Payload**

9913 D.5.2.3.4.2 Payload Details

9914 **Implementation Date/Time (mandatory):** A UTC Time field to indicate the date/time at which
9915 all existing display messages should be cleared.

9916

9917 ## D.5.3 Client

9918 ### D.5.3.1 Dependencies

9919 Support for ZCL Data Types.

9920 No dependencies exist for other Smart Energy Clusters.

9921 ### D.5.3.2 Attributes

9922 None.

9923 ### D.5.3.3 Commands Generated

9924 The command IDs generated by the Messaging cluster are listed in Table D-119.

9925 **Table D-119– Messaging Client Commands**

Command Identifier Field Value	**Description**	**Mandatory / Optional**
0x00	*Get Last Message*	M
0x01	*Message Confirmation*	M
0x02	*GetMessageCancellation*	O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x03 – 0xff	Reserved	

9926
9927

9928 D.5.3.3.1 **Get Last Message Command**

9929 This command has no payload.

9930 D.5.3.3.1.1 Effect on Receipt

9931 On receipt of this command, the device shall send a *Display Message* or *Display Protected*
9932 *Message* command as appropriate (refer to sub-clauses D.5.2.3.1 and D.5.2.3.3). A ZCL Default
9933 Response with status NOT_FOUND shall be returned if no message is available.

9934

9935 D.5.3.3.2 **Message Confirmation Command**

9936 The *Message Confirmation* command described in Figure D-96 provides an indication that a
9937 Utility Customer has acknowledged and/or accepted the contents of a previously sent message.
9938 Enhanced Message Confirmation commands shall contain an answer of 'NO', 'YES' and/or a
9939 message confirmation string.

9940 If the optional *Message Confirmation Response* is required, the *Message Confirmation Control*
9941 field shall also be present.

Octets	**4**	**4**	**1**	**1-21**
Data Type	Unsigned 32-bit integer	UTCTime	8-bit BitMap	Octet String
Field Name	Message ID (M)	Confirmation Time (M)	Message Confirmation Control (O)	Message Confirmation Response (O)

9942 **Figure D-96– *Message Confirmation* Command Payload**

9943 D.5.3.3.2.1 Payload Details

9944 **Message ID (mandatory):** A unique unsigned 32-bit number identifier for the message being
9945 confirmed.

9946 **Confirmation Time (mandatory):** UTCTime of user confirmation of message.

9947 **Message Confirmation Control (optional)**: An 8-bit BitMap field indicating the simple
9948 confirmation that is contained within the response. Bit encoding of this field is outlined in Table
9949 D-120; if this optional field is not available, a default value of 0x00 shall be used.

9950

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

9951 **Table D-120– Message Confirmation Control**

Bit	Enumeration	Value	Description
0	'NO' Returned		The answer is 'NO'
1	'YES' Returned		The answer is 'YES'
Bits 2 - 7	Reserved		

9952

9953 **Message Confirmation Response (optional)**: A ZCL Octet String containing the message to be
9954 returned. The first Octet indicates length. The string shall be encoded in the UTF-8 format. If this
9955 optional field is not available, a default value of 0x00 shall be used.

9956

9957 D.5.3.3.3 **GetMessageCancellation Command**

9958 ***Note:*** *The GetMessageCancellation command in this revision of this specification is provisional*
9959 *and not certifiable. This feature may change before reaching certifiable status in a future revision*
9960 *of this specification.*

9961 This command initiates the return of the first (and maybe only) *Cancel All Messages* command
9962 held on the associated server, and which has an implementation time equal to or later than the
9963 value indicated in the payload.

9964 D.5.3.3.3.1 Payload Format

Octets	4
Data Type	UTC Time
Field Name	Earliest Implementation Time (M)

9965 **Figure D-97– Get*MessageCancellation* Command Payload**

9966 D.5.3.3.3.2 Payload Details

9967 **Earliest Implementation Time (mandatory):** UTC Timestamp indicating the earliest
9968 implementation time of a *Cancel All Messages* command to be returned.

9969 D.5.3.3.3.3 When Generated

9970 This command is generated when the client device wishes to fetch any pending *Cancel All*
9971 *Messages* command from the server (see D.5.2.3.4 for further details). In the case of a BOMD,
9972 this may be as a result of the associated Notification flag.

9973 A ZCL Default response with status NOT_FOUND shall be returned if there is no *Cancel All*
9974 *Messages* command available that satisfies the requested criteria.

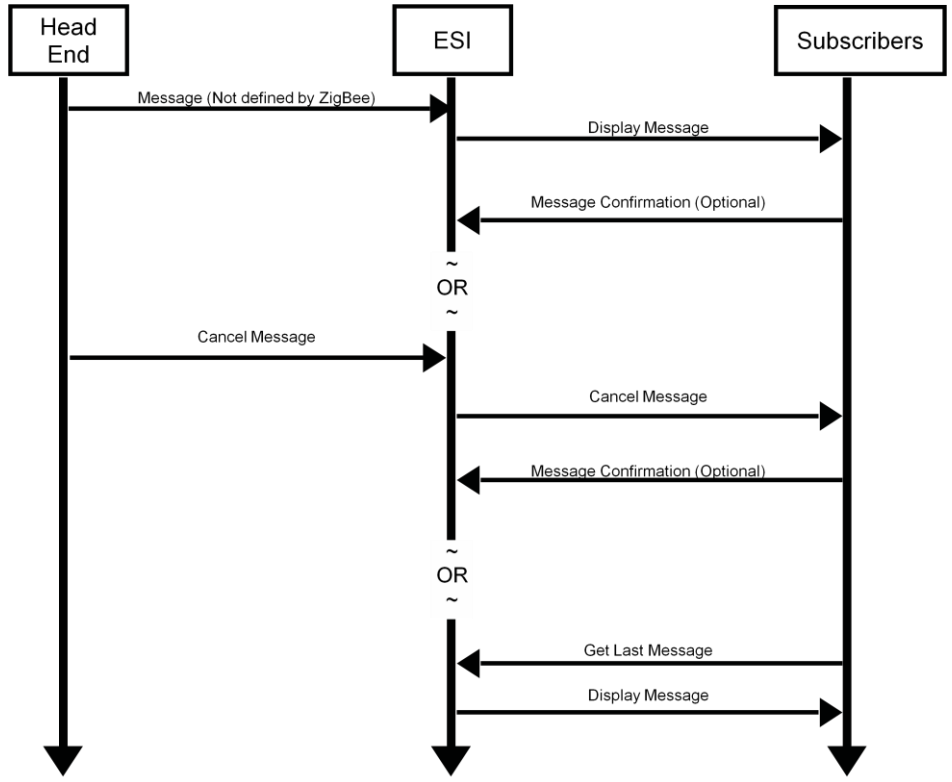
Copyright © 2014 ZigBee Standards Organization. All rights reserved.

9975

9976 ### D.5.4 Application Guidelines

9977 For Server and Client transactions, please refer to Figure D-98.

9978



9979

9980 **Figure D-98– Client/Server Message Command Exchanges**

9981
9982

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

D.6 Tunneling Cluster

Note: *The optional support for flow control within the cluster in this revision of this specification is provisionary and not certifiable. This feature set may change before reaching certifiable status in a future revision of this specification.*

D.6.1 Overview

The tunneling cluster provides an interface for tunneling protocols. It is comprised of commands and attributes required to transport any existing metering communication protocol within the payload of standard ZigBee frames (including the handling of issues such as addressing, fragmentation and flow control). Examples for such protocols are DLMS/COSEM, IEC61107, ANSI C12, M-Bus, ClimateTalk etc.

The tunneling foresees the roles of a server and a client taking part in the data exchange. Their roles are defined as follows:

- **Client:** Requests a tunnel from the server and closes the tunnel if it is no longer needed.
- **Server:** Provides and manages tunnels to the clients.

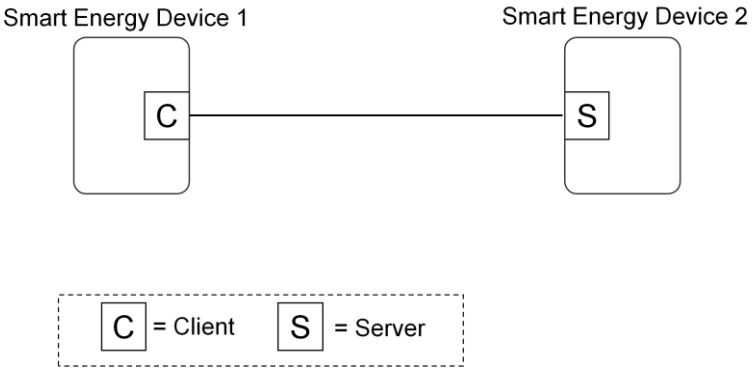


Figure D-99– A Client Requests a Tunnel From a Server to Exchange Complex Data in Both Directions

The data exchange through the tunnel is symmetric. This means both client and server provide the commands to transfer data (*TransferData*). And both must make sure that only the partner to which the tunnel has been built up is granted read/write access to it (e.g. tunnel identifier protection through checking the MAC address).

Sleepy devices either close the tunnel immediately after they have pushed their data through it, or leave it open in which case an attribute in the server (*CloseTunnelTimeout*) decides whether the tunnel is closed from the server side during the sleeping phase or not. It is recommended that battery-powered (sleepy) devices fulfil the role of the Tunneling cluster client (and therefore have control over when they request a tunnel from the server).

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

If data is transferred to a non-existent or wrong tunnel identifier, the receiver generates an error message (T*ransferDataError*).

The server may support more than one tunneling protocol. The type of tunnel to be opened is a mandatory parameter (*ProtocolID*) of the tunnel request (*RequestTunnel*) that the client needs to send to the server in order to set up a new tunnel. The response from the server (*RequestTunnelResponse*) will contain a parameter with the status of the tunnel (*TunnelStatus*). If the tunnel request was successful, a unique identifier (*TunnelID*) is returned within the response. In an error case (e.g. the requested protocol is not supported) the status contains the type of error. The optional *GetSupportedTunnelProtocols* command provides a way to read out the supported protocols from the server. If the *GetSupportedTunnelProtocols* command is not supported then either the client knows the supported protocols a priori or it has to try several times using different *ProtocolIDs* until the server responds with the tunnel status *Success*.

The tunneling cluster adds optional support for flow control to handle streaming protocols such as IEC61107. If implemented, flow control messages are provided to control the data flow and send acknowledges to data messages on application level. However, flow control is an optional feature and disabled per default. In the default case, the acknowledge messages (*AckTransferData*) must not be sent in order to reduce complexity and prevent from unneeded overhead.

The following sequence describes a typical usage:

1 The client issues a service discovery to find devices which support the tunneling server cluster. The discovery may either be directed to one device, if its address is known, or be a broadcast (*MatchSimpleDescriptor*).

2 The response to the discovery from the server contains an endpoint number (*SimpleDescriptor*). Using this endpoint, the client directs a tunnel request to a given server. Together with the request, the client is required to provide an enumeration with the ID of the protocol that shall be tunneled. There is the possibility to request tunnels for manufacturer specific protocols. In this case, the *ProtocolID* has to be followed by a *ZigBee ManufacturerCode* to open the tunnel. An additional parameter for *FlowControlSupport* accompanies the request, together with an indication of the client's incoming buffer size (*RequestTunnel* (*ProtocolID, ManufacturerCode, FlowControlSupport, MaximumIncomingTransferSize*)).

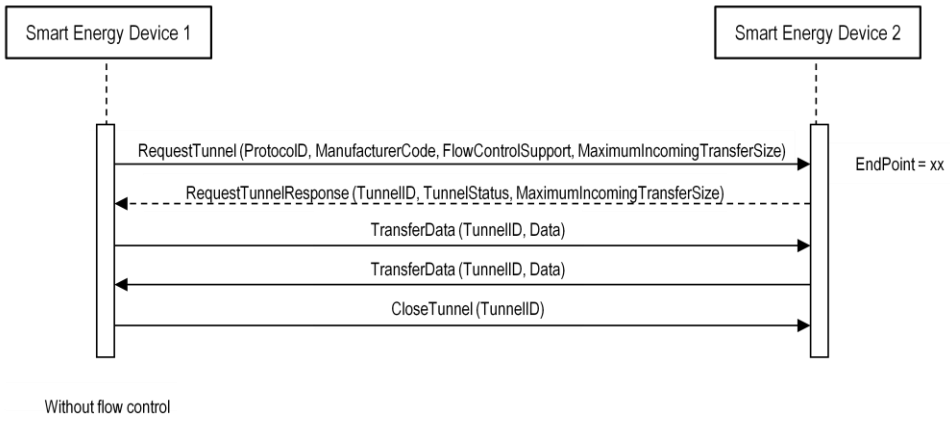
3 If the server supports the protocol, it allocates the required resources, assigns a tunnel identifier and returns the ID number within the response including an additional tunnel status that the command was successful and the server's incoming buffer size. If the command failed, the status contains the reason in form of an error code (*RequestTunnelResponse* (*TunnelID, TunnelStatus, MaximumIncomingTransferSize*)). The tunnel identifier number would then be invalid in this case.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

10047 **4** Both server and client may exchange data (*TransferData(Data)*). In case the optional flow
10048 control is utilized, each data transfer is acknowledged
10049 (*AckTransferData(NumberOfOctetsLeft)*). Additionally, there is the possibility to stop
10050 (*AckTransferData*(0)) and resume (*ReadyData(NumberOfOctetsLeft)* the data transfer.

10051 **5** After the transfer has been successfully completed, the client closes the tunnel again freeing
10052 the tunnel identifier in the server (*CloseTunnel(TunnelID)*). If not, the server closes the
10053 tunnel by itself after *CloseTunnelTimeout* seconds.

10054 The following sequence diagrams show the client/server model and the typical usage of the
10055 cluster without (Figure D-100) and with (Figure D-101) flow control.
10056



10057

10058 **Figure D-100– SE Device 1 (Client) Requests a Tunnel From SE Device 2 (Server) to Transfer Data Without**
10059 **Flow Control (Default)[17]**
10060
10061

[17] CCB 1939

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

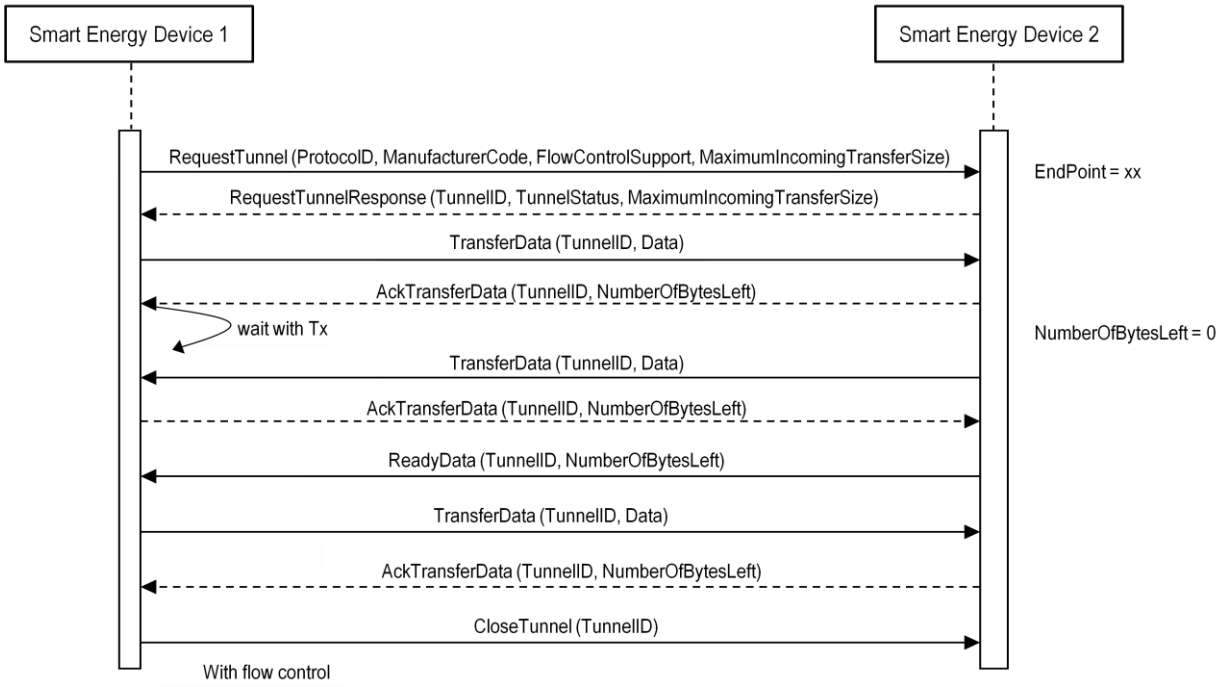


Figure D-101– SE Device 1 (Client) Requests a Tunnel From SE Device 2 (Server) to Transfer Data With Flow Control[18]

D.6.2 Server

D.6.2.1 Dependencies

This cluster requires APS fragmentation [B3] to be implemented, with maximum transfer sizes defined by the device's negotiated input buffer sizes.

D.6.2.2 Attributes

Table D-121– Tunneling Server Attributes

Identifier	Name	Type	Range	Access	Default	Man. /Opt.
0x0000	*CloseTunnelTimeout*	Unsigned 16-bit Integer	0x0001-0xFFFF	Read Only	0xFFFF	M

[18] CCB 1939

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

10074 D.6.2.2.1 **CloseTunnelTimeout Attribute**

10075 *CloseTunnelTimeout* defines the minimum number of seconds that the server waits on an
10076 inactive tunnel before closing it on its own and freeing its resources (without waiting for the
10077 *CloseTunnel* command from the client). Inactive means here that the timer is re-started with
10078 each new reception of a command.0x0000 is an invalid value.

10079

10080 ### D.6.2.3 Parameters

10081 The table below contains a summary of all parameters passed to or returned by the server
10082 commands. These values are considered as parameters (and not attributes) in order to facilitate
10083 the handling of the tunneling cluster for both the client and the server side. The parameters
10084 cannot be read or written via ZCL global commands. The detailed description of these
10085 parameters can be found in the according command sections of the document.

10086 **Table D-122– Cluster Parameters Passed Through Commands**

Name	Type	Range	Default	Mandatory / Optional
ProtocolID	8-bit enumeration	0x01 – 0xFF	0x00	M
ManufacturerCode	Unsigned 16-bit integer	0x0000 – 0xFFFF	0x00	M
FlowControlSupport	Boolean	TRUE or FALSE	FALSE	M
MaximumIncoming TransferSize	Unsigned 16-bit integer	0x0000 – 0xFFFF	1500	M
TunnelID	Unsigned 16-bit integer	0x0000 – 0xFFFF	(Return value)	M
Data	Octet string	-	-	M
NumberOfOctetsLeft	Unsigned 16-bit integer	0x0000 – 0xFFFF	-	M
TunnelStatus	Unsigned 8-bit integer	0x00 – 0x04	-	M
TransferDataStatus	Unsigned 8-bit integer	0x00 – 0x01	-	M

10087
10088
10089
10090

10091 ### D.6.2.4 Commands Received

10092 Table D-123 lists cluster-specific commands received by the server.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

10093 **Table D-123– Cluster -specific Commands Received by the Server**

Command Identifier FieldValue	Description	Mandatory/ Optional
0x00	*RequestTunnel*	M
0x01	*CloseTunnel*	M
0x02	*TransferData*	M
0x03	*TransferDataError*	M
0x04	*AckTransferData*	O
0x05	*ReadyData*	O
0x06	*GetSupportedTunnelProtocols*	O

10094
10095

10096 D.6.2.4.1 **RequestTunnel Command**

10097 *RequestTunnel* is the client command used to setup a tunnel association with the server. The
10098 request payload specifies the protocol identifier for the requested tunnel, a manufacturer code
10099 in case of proprietary protocols and the use of flow control for streaming protocols.

10100 D.6.2.4.1.1 Payload Format

Octets	1	2	1	2
Data Type	8-bit enumeration	Unsigned 16-bit integer	Boolean	Unsigned 16-bit integer
Field Name	ProtocolID (M)	Manufacturer Code (M)	FlowControl Support (M)	Maximum Incoming TransferSize (M)

10101 **Figure D-102– *RequestTunnel* Command Payload**
10102

10103 D.6.2.4.1.2 Payload Details

10104 **ProtocolID:** An enumeration representing the identifier of the metering communication protocol
10105 for which the tunnel is requested. Table D-124 lists the possible values for the *ProtocolID*.
10106 The values above 199 may be used for manufacturer specific protocols.

10107

10108

10109 **Table D-124– ProtocolID Enumerations**

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Values	Description
0	DLMS/COSEM (IEC 62056)
1	IEC 61107
2	ANSI C12
3	M-BUS
4	SML
5	ClimateTalk
6	GB-HRGP
7 to 199	Reserved for future growth
200 to 254	Manufacturer-defined protocols
255	Reserved

Manufacturer Code: A code that is allocated by the ZigBee Alliance, relating the manufacturer to a device and – for the tunneling - a manufacturer specific protocol. The parameter is ignored when the *ProtocolID* value is less than 200. This allows for 55 manufacturer-defined protocols for each manufacturer to be defined. A value of 0xFFFF indicates that the Manufacturer Code is not used.

FlowControlSupport: A boolean type parameter that indicates whether flow control support is requested from the tunnel (TRUE) or not (FALSE). The default value is FALSE (no flow control).

MaximumIncomingTransferSize: A value that defines the size, in octets, of the maximum data packet that can be transferred to the client in the payload of a single *TransferData* command.

D.6.2.4.1.3 When Generated

Is never generated by the server.

D.6.2.4.1.4 Effect on Receipt

Triggers a process within the server to allocate resources and build up a new tunnel. A *RequestTunnelResponse* is generated and sent back to the client containing the result of the *RequestTunnel* command.

D.6.2.4.2 CloseTunnel Command

Client command used to close the tunnel with the server. The parameter in the payload specifies the tunnel identifier of the tunnel that has to be closed. The server leaves the tunnel open and the assigned resources allocated until the client sends the *CloseTunnel* command or the *CloseTunnelTimeout* fires.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

D.6.2.4.2.1 Payload Format

Octets	2
Data Type	Unsigned 16-bit integer
Field Name	TunnelID (M)

Figure D-103– *CloseTunnel* Command Payload

D.6.2.4.2.2 Payload Details

TunnelID: The identifier of the tunnel that shall be closed. It is the same number that has been previously returned in the response to a *RequestTunnel* command. Valid numbers range between 0..65535 and must correspond to a tunnel that is still active and maintained by the server.

D.6.2.4.2.3 When Generated

This command is never generated by the server.

D.6.2.4.2.4 Effect on Receipt

In case the given *TunnelID* is correct, the server closes the tunnel and frees the resources. The associated tunnel is no longer maintained. If the *TunnelID* value does not match an active tunnel on the server, the server shall return a ZCL Default Response with status NOT_FOUND.

D.6.2.4.3 **TransferData Command**

Command that indicates (if received) that the client has sent data to the server. The data itself is contained within the payload.

D.6.2.4.3.1 Payload Format

Octets	2	Variable
Data Type	Unsigned 16-bit integer	Octets
Field Name	TunnelID (M)	Data (M)

Figure D-104– *TransferData* Command Payload

D.6.2.4.3.2 Payload Details

TunnelID: A number between 0..65535 that uniquely identifies the tunnel that has been allocated in the server triggered through the *RequestTunnel* command. This ID must be used to send data through the tunnel or passed with any commands concerning that specific tunnel.

Data: Octet containing the data to be transferred through the tunnel in the format of the communication protocol for which the tunnel has been requested and opened. The payload

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

10156 contains the assembled data exactly as it was sent by the client. Theoretically, its length is
10157 solely limited through the fragmentation algorithm and the RX/TX transfer buffer sizes within the
10158 communication partners. The content of the payload is up to the application sending the data. It
10159 is neither guaranteed, that it contains a complete PDU nor is any other assumption on its
10160 internal format made. This is left up to the implementer of the specific protocol tunnel
10161 behavior.

10162 D.6.2.4.3.3 When Generated

10163 Is generated whenever the server wants to tunnel protocol data to the client.

10164 D.6.2.4.3.4 Effect on Receipt

10165 Indicates that the server has received tunneled protocol data from the client.

10166 D.6.2.4.4 **TransferDataError Command**

10167 This command is generated by the receiver of a *TransferData* command if the tunnel status
10168 indicates that something is wrong. There are three cases in which *TransferDataError* is sent:

10169 • The *TransferData* received contains a *TunnelID* that does not match to any of the active
10170 tunnels of the receiving device. This could happen if a (sleeping) device sends a
10171 *TransferData* command to a tunnel that has been closed by the server after the
10172 *CloseTunnelTimeout*.

10173 • The *TransferData* received contains a proper *TunnelID* of an active tunnel, but the device
10174 sending the data does not match to it.

10175 • The *TransferData* received contains more data than indicated by the
10176 *MaximumIncomingTransferSize* of the receiving device.

10177 D.6.2.4.4.1 Payload Format

Octets	2	1
Data Type	Unsigned 16-bit integer	Unsigned 8-bit integer
Field Name	TunnelID (M)	TransferDataStatus (M)

10178 **Figure D-105– *TransferDataError* Command Payload**

10179 D.6.2.4.4.2 Payload Details

10180 **TunnelID:** A number between 0..65535 that uniquely identifies the tunnel that has been
10181 allocated in the server triggered through the *RequestTunnel* command. This ID must be used
10182 for the data transfer through the tunnel or passed with any commands concerning that specific
10183 tunnel.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

TransferDataStatus: The *TransferDataStatus* parameter indicates the error that occurred within the receiver after the last *TransferData* command.

The *TransferDataStatus* values are shown in Table D-125.

Table D-125– TransferDataStatus Values

Value	Description	Remarks
0x00	No such tunnel	The *TransferData* command contains a TunnelID of a non-existent tunnel.
0x01	Wrong device	The *TransferData* command contains a TunnelID that does not match the device sending the data.
0x02	Data overflow	The *TransferData* command contains more data than indicated by the *MaximumIncomingTransferSize* of the receiving device
0x03 – 0xFF	Reserved	Should not be returned and indicates an unknown error.

D.6.2.4.4.3 When Generated

Is generated if the server wants to tell the client that there was something wrong with the last *TransferData* command.

D.6.2.4.4.4 Effect on Receipt

Indicates that the client wants to tell the server that there was something wrong with the last *TransferData* command.

D.6.2.4.5 **AckTransferData Command**

Command sent in response to each *TransferData* command in case – and only in case – flow control has been requested by the client in the *TunnelRequest* command and is supported by both tunnel endpoints. The response payload indicates the number of octets that may still be received by the receiver.

D.6.2.4.5.1 Payload Format

Octets	2	2
Data Type	Unsigned 16-bit integer	Unsigned 16-bit Integer
Field Name	TunnelID (M)	NumberOfBytes Left (M)

Figure D-106– *AckTransferData* Command Payload

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

10202 D.6.2.4.5.2 Payload Details

10203 **TunnelID:** A number between 0..65535 that uniquely identifies the tunnel that has been
10204 allocated in the server triggered through the *RequestTunnel* command. This ID must be used
10205 for the data transfer through the tunnel or passed with any commands concerning that specific
10206 tunnel.

10207 **NumberOfBytesLeft:** Indicates the number of bytes that may still be received by the initiator of
10208 this command (receiver). It is most likely the remaining size of the buffer holding the data that is
10209 sent over *TransferData*. As an example: A value of 150 indicates that the next *TransferData*
10210 command must not contain more than 150 bytes of payload or data will get lost. A value of 0
10211 indicates that there is no more space left in the receiver and the sender should completely stop
10212 sending data. After the reception of a *ReadyData* command, the sender may continue its data
10213 transfer.

10214 D.6.2.4.5.3 When Generated

10215 If flow control is on, the command is issued by the server to inform the client that the last
10216 *TransferData* command has been successfully received and how much space is left to receive
10217 further data.

10218 D.6.2.4.5.4 Effect on Receipt

10219 If flow control is on, the reception of this command indicates that the client wants to inform the
10220 server that the last *TransferData* command has been successfully received and how much space
10221 is left to receive further data.

10222 D.6.2.4.6 **ReadyData Command**

10223 The *ReadyData* command is generated – after a receiver had to stop the dataflow using the
10224 *AckTransferData(0)* command – to indicate that the device is now ready to continue receiving
10225 data. The parameter *NumberOfOctetsLeft* gives a hint on how much space is left for the next
10226 data transfer. The *ReadyData* command is only issued if flow control is enabled.

10227 D.6.2.4.6.1 Payload Format

Octets	2	2
Data Type	Unsigned 16-bit integer	Unsigned 16-bit Integer
Field Name	TunnelID (M)	NumberOfOctets Left (M)

10228 **Figure D-107– *ReadyData* Command Payload**

10229 D.6.2.4.6.2 Payload Details

10230 **TunnelID:** A number between 0..65535 that uniquely identifies the tunnel that has been
10231 allocated in the server triggered through the *RequestTunnel* command. This ID must be used

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

for the data transfer through the tunnel or passed with any commands concerning that specific tunnel.

NumberOfOctetsLeft: Indicates the number of octets that may be received by the initiator of this command (receiver). It is most likely the remaining size of the buffer holding the data that is sent over *TransferData*. As an example: A value of 150 indicates that the next *TransferData* command must not contain more than 150 bytes of payload or data will get lost. The value must be larger than 0. As for its exact value, it is up to the implementer of the cluster to decide what flow control algorithm shall be applied.

D.6.2.4.6.3 When Generated

If generated by the server, this command informs the client that it may now continue to send and how much space is left within the server to receive further data.

D.6.2.4.6.4 Effect on Receipt

If received by the server, this command informs the server that it may now continue to send and how much space is left within the client to receive further data.

D.6.2.4.7 Get Supported Tunnel Protocols Command

Get Supported Tunnel Protocols is the client command used to determine the tunnel protocols supported on another device.

D.6.2.4.7.1 Payload Format

Octets	1
Data Type	Unsigned 8-bit Integer
Field Name	Protocol Offset

Figure D-108– *Get Supported Tunnel Protocols* Command Payload

D.6.2.4.7.2 Payload Details

Protocol Offset: Where there are more protocols supported than can be returned in a single *Supported Tunnel Protocols Response* command, this field allows an offset to be specified on subsequent *Get Supported Tunnel Protocols* commands. An offset of zero (0x00) should be used for an initial (or only) *Get Supported Tunnel Protocols* command (indicating that the returned list of protocols should commence with first available protocol). As a further example, if 10 protocols had previously been returned, the next *Get Supported Tunnel Protocols* command should use an offset of 10 (0x0A) to indicate the 11th available protocol should be the first returned in the next response.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

10260 D.6.2.4.7.3 Effect on Receipt

10261 On receipt of this command, a device will respond with a *Supported Tunnel Protocols*
10262 *Response* command, indicating the tunnel protocols it supports (see sub- clause D.6.2.5.6 for
10263 further details).

10264 ## D.6.2.5 Commands Generated

10265 Table D-126 lists commands that are generated by the server.

10266 **Table D-126– Cluster-Specific Commands Sent by the Server**

Command Identifier FieldValue	Description	Mandatory/ Optional
0x00	*RequestTunnelResponse*	M
0x01	*TransferData*	M
0x02	*TransferDataError*	M
0x03	*AckTransferData*	O
0x04	*ReadyData*	O
0x05	*Supported Tunnel Protocols Response*	O
0x06	*TunnelClosureNotification*	O

10267
10268

10269 D.6.2.5.1 **RequestTunnelResponse Command**

10270 *RequestTunnelResponse* is sent by the server in response to a *RequestTunnel* command
10271 previously received from the client. The response contains the status of the *RequestTunnel*
10272 command and a tunnel identifier corresponding to the tunnel that has been set-up in the server
10273 in case of success.

10274 D.6.2.5.1.1 Payload Format

Octets	2	1	2
Data Type	Unsigned 16-bit Integer	Unsigned 8-bit Integer	Unsigned 16-bit Integer
Field Name	TunnelID (M)	TunnelStatus (M)	Maximum Incoming TransferSize (M)

10275 **Figure D-109– *RequestTunnelResponse* Command Payload**

10276 D.6.2.5.1.2 Payload Details

10277 **TunnelID:** A number between 0..65535 that uniquely identifies the tunnel that has been
10278 allocated in the server triggered through the *RequestTunnel* command. This ID must now be

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

used to send data through this tunnel (*TunnelID*, *TransferData*) and is also required to close the tunnel again (*CloseTunnel*). If the command has failed, the *TunnelStatus* contains the reason of the error and the *TunnelID* is set to 0xFFFF.

TunnelStatus: The *TunnelStatus* parameter indicates the server's internal status after the execution of a *RequestTunnel* command.

The *TunnelStatus* values are shown in Table D-127.

Table D-127– TunnelStatus Values

Value	Description	Remarks
0x00	Success	The tunnel has been opened and may now be used to transfer data in both directions.
0x01	Busy	The server is busy and cannot create a new tunnel at the moment. The client may try again after a recommended timeout of 3 minutes.
0x02	No more tunnel IDs	The server has no more resources to setup requested tunnel. Clients should close any open tunnels before retrying.
0x03	Protocol not supported	The server does not support the protocol that has been requested in the ProtocolID parameter of the *RequestTunnel* command.
0x04	Flow control not supported	Flow control has been requested by the client in the *RequestTunnel* command but cannot be provided by the server (missing resources or no support).
0x05 to 0xFF	Reserved	Should not be returned and indicates an unknown error.

MaximumIncomingTransferSize: A value that defines the size, in octets, of the maximum data packet that can be transferred to the server in the payload of a single *TransferData* command.

D.6.2.5.1.3 When Generated

Is generated in reply to a *RequestTunnel* command to inform the client about the result of the request.

D.6.2.5.1.4 Effect on Receipt

Should never be received by the server.

D.6.2.5.2 TransferData Command

Command that transfers data from server to the client. The data itself has to be placed within the payload.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

10297 D.6.2.5.2.1 Payload Format

Octets	2	Variable
Data Type	Unsigned 16-bit integer	Octets
Field Name	TunnelID (M)	Data (M)

10298 **Figure D-110– *TransferData* Command Payload**

10299 D.6.2.5.2.2 Payload Details

10300 **TunnelID:** A number between 0..65535 that uniquely identifies the tunnel that has been
10301 allocated in the server triggered through the *RequestTunnel* command. This ID must be used
10302 for the data transfer through the tunnel or passed with any commands concerning that specific
10303 tunnel.

10304 **Data:** Octets containing the data to be transferred through the tunnel in the format of the
10305 communication protocol for which the tunnel has been requested and opened. The payload
10306 containing the assembled data exactly as it has been sent away by the client. Theoretically, its
10307 length is solely limited through the fragmentation algorithm and the RX/TX transfer buffer sizes
10308 within the communication partners. The content of the payload is up to the application
10309 sending the data. It is not guaranteed that it contains a complete PDU, nor is any assumption to
10310 be made on its internal format (which is left up to the implementer of the specific tunnel
10311 protocol).

10312 D.6.2.5.2.3 When Generated

10313 Is generated when the server wants to tunnel protocol data to the client.

10314 D.6.2.5.2.4 Effect on Receipt

10315 Indicates that the server has received tunneled protocol data from the client.

10316 D.6.2.5.3 **TransferDataError Command**

10317 See sub-clause D.6.2.4.4.

10318 D.6.2.5.4 **AckTransferData Command**

10319 See sub-clause D.6.2.4.5.

10320 D.6.2.5.5 **ReadyData Command**

10321 See sub-clause D.6.2.4.6.

10322 D.6.2.5.6 **Supported Tunnel Protocols Response Command**

10323 *Supported Tunnel Protocols Response* is sent in response to a *Get Supported Tunnel*
10324 *Protocols* command previously received. The response contains a list of tunnel protocols

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

supported by the device; the payload of the response should be capable of holding up to 16 protocols.

D.6.2.5.6.1 Payload Format

Octets	1	1	3	…	3
Data Type	Boolean	Unsigned 8-bit Integer			
Field Name	Protocol List Complete	Protocol Count	Protocol 1	…	Protocol n

Figure D-111– *Supported Tunnel Protocols Response* Command Payload

where each *Protocol* field shall be formatted as:

Octets	2	1
Data Type	Unsigned 16-bit Integer	8-bit Enumeration
Field Name	Manufacturer Code	Protocol ID

Figure D-112– *Supported Tunnel Protocols Response* Command Protocol Fields

D.6.2.5.6.2 Payload Details

Protocol List Complete: The Protocol List Complete field is a Boolean; a value of 0 indicates that there are more supported protocols available (if more than 16 protocols are supported). A value of 1 indicates that the list of supported protocols is complete.

Protocol Count: The number of Protocol fields contained in the response.

Manufacturer Code: A code that is allocated by the ZigBee Alliance, relating the manufacturer to a device and - for tunneling - a manufacturer specific protocol. A value of 0xFFFF indicates a standard (i.e. non- manufacturer specific) protocol

Protocol ID: An enumeration representing the identifier of the metering communication protocol for the supported tunnel. Table D-124 lists the possible values for standard protocols

D.6.2.5.6.3 When Generated

Is generated in reply to a *Get Supported Tunnel Protocols* command. to indicate the tunnel protocols supported by the device

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

10345 D.6.2.5.7 **TunnelClosureNotification Command**

10346 *TunnelClosureNotification* is sent by the server to indicate that a tunnel has been closed due to
10347 expiration of a *CloseTunnelTimeout*.

10348

10349 D.6.2.5.7.1 Payload Format

Octets	2
Data Type	Unsigned 16-bit Integer
Field Name	TunnelID (M)

10350 **Figure D-113– *TunnelClosureNotification* Command Payload**

10351 D.6.2.5.7.2 Payload Details

10352 **TunnelID:** The identifier of the tunnel that has been closed. It is the same number that has been
10353 previously returned in the response to a *RequestTunnel* command. Valid numbers range
10354 between 0..65535 and must correspond to a tunnel that was still active and maintained by the
10355 server.

10356 D.6.2.5.7.3 When Generated

10357 The command is sent by a server when a tunnel is closed due to expiration of
10358 *CloseTunnelTimeout*. It is sent unicast to the client that had originally requested that tunnel.

10359

10360 D.6.3 Client

10361 D.6.3.1 Dependencies

10362 This cluster requires APS fragmentation [B3] to be implemented, with maximum transfer sizes
10363 defined by the device's negotiated input buffer sizes.

10364 D.6.3.2 Attributes

10365 The client has no attributes.

10366 D.6.3.3 Commands Received

10367 The client receives the cluster-specific response commands detailed in D.6.2.5.

10368 D.6.3.4 Commands Generated

10369 The client generates the cluster-specific commands detailed in D.6.2.4, as required by the
10370 application.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

10371

10372

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

10373 ## D.7 Prepayment Cluster

10374 ### D.7.1 Overview

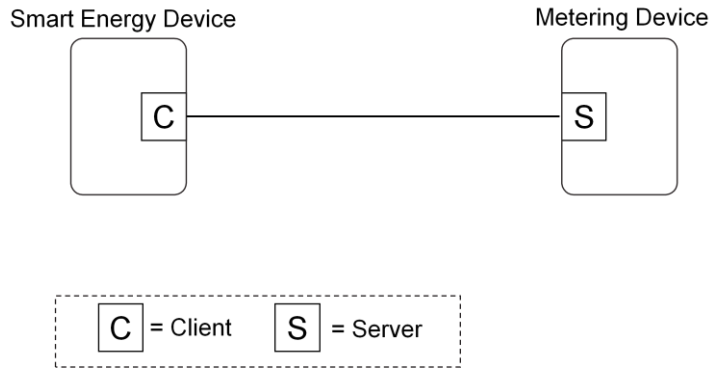
10375 The Prepayment Cluster provides the facility to pass messages relating to the accounting
10376 functionality of a meter between devices on the HAN. It allows for the implementation of a
10377 system conforming to the set of standards relating to Payment Electricity Meters (IEC 62055)
10378 and also for the case where the accounting function is remote from the meter. Prepayment is
10379 used in situations where the supply of a service may be interrupted or enabled under the control
10380 of the meter or system in relation to a payment tariff. The accounting process may be within
10381 the meter or elsewhere in the system. The amount of available credit is decremented as the
10382 service is consumed and is incremented through payments made by the consumer. Such a
10383 system allows the consumer to better manage their energy consumption and reduces the risk
10384 of bad debt owing to the supplier.

10385 In the case where the accounting process resides within the meter, credit updates are sent to the
10386 meter from the ESI. Such messages are out of scope of this cluster. The cluster allows credit
10387 status to be made available to other devices on the HAN for example to enable the
10388 consumers to view their status on an IHD. It also allows them to select emergency credit if
10389 running low and also, where local markets allow, restoring their supply remotely from within the
10390 HAN.

10391 In the case where the accounting process resides in the head end (Central Wallet scheme), the
10392 metering system provides usage information to the head end for it to calculate the state of
10393 available credit in the consumer's account. The head end will pass down to the metering system
10394 data that will be of use to the consumer, for distribution on the HAN. The head end will also
10395 send commands to interrupt or restore the supply depending on the state of the account.

10396 In either case, there will be the need to display credit status and this may be in monetary terms
10397 or in energy terms. If running in monetary mode, the units of measure will be defined in the
10398 Price Cluster, if in energy terms, the unit of measure will be defined in the Metering Cluster.

10399



10400

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

10401 **Figure D-114– Prepay Cluster Client Server Example**

10402

10403 ## D.7.2 Server

10404 ### D.7.2.1 Dependencies

10405 • Support for ZCL Data Types

10406 • Events carried using this cluster include a timestamp with the assumption that target devices
10407 maintain a real time clock. Devices can acquire and synchronize their internal clocks via the
10408 ZCL Time server.

10409 • Use of the Price cluster is Mandatory when using the Prepayment cluster in Currency mode.

10410 • The Calendar cluster shall be used to set up the Friendly Credit period that the prepayment
10411 meter shall use (see Annex D.9 for further details).

10412 • Use of the Metering cluster is Mandatory when using the Prepayment cluster in any mode.

10413 • Use of the Device Management cluster is mandatory when using the disconnection function
10414 within the Prepayment cluster.

10415 ### D.7.2.2 Attributes

10416 For convenience, the attributes defined in this specification are arranged into sets of related
10417 attributes; each set can contain up to 256 attributes. Attribute identifiers are encoded such that the
10418 most significant Octet specifies the attribute set and the least significant Octet specifies the
10419 attribute within the set. The currently defined attribute sets are listed in the following Table
10420 D-128.

10421 **Table D-128– Prepayment Attribute Sets**

Attribute Set Identifier	Description
0x00	Prepayment Information Set
0x01	Top-up Attribute Set
0x02	Debt Attribute Set
0x03	Reserved
0x04	Alarms Set
0x05	Historical Cost Consumption Information Set
0x06 – 0xFF	Reserved

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

10422

10423 D.7.2.2.1 **Prepayment Information Attribute Set**

10424 The following set of attributes provides access to the standard information relating to a
10425 Prepayment meter.

10426 **Table D-129– P*repayment Information* Attribute Set**

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0000	*Payment Control Configuration*	16-bit Bitmap	0x0000 to 0xFFFF	Read only	0x0000	M
0x0001	*Credit Remaining*	Signed 32-bit Integer	- 0x7FFFFFFF To +0x7FFFFFFF	Read only	-	O
0x0002	*Emergency Credit Remaining*	Signed 32-bit Integer	-0x7FFFFFFF to +0x7FFFFFFF	Read only	-	O
0x0003	*Credit Status*	8-bit Bitmap	0x00 to 0x40	Read only	0x00	O
0x0004	CreditRemaining TimeStamp	UTCTime		Read only	-	O
0x0005	Accumulated Debt	Signed 32-bit Integer	-0x7FFFFFFF To +0x7FFFFFFF	Read only	-	O
0x0006	OverallDebtCap	Signed 32-bit Integer	-0x7FFFFFFF To +0x7FFFFFFF	Read only	-	O
0x0007 – 0x000F	Reserved					
0x0010	EmergencyCredit Limit/Allowance	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x0011	EmergencyCredit Threshold	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x0012 – 0x001F	Reserved					
0x0020	TotalCreditAdded	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0021	MaxCreditLimit	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read only	-	O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x0022	MaxCredit PerTopUp	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x0023 – 0x002F	Reserved					
0x0030	FriendlyCredit Warning	Unsigned 8-bit Integer	0x00 to 0xFF	Read only	0x0A	O
0x0031	LowCredit Warning	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x0032	IHDLow CreditWarning	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read/ Write	-	O
0x0033	InterruptSuspend Time	Unsigned 8-bit Integer	0x00 to 0xFF	Read only	60	O
0x0034	RemainingFriendly CreditTime	Unsigned 16-bit Integer	0x0000 to 0xFFFF	Read only	-	O
0x0035	NextFriendly CreditPeriod	UTCTime		Read only	-	O
0x0036 – 0x003F	Reserved					
0x0040	CutOffValue	Signed 32-bit Integer	-0x7FFFFFFF to +0x7FFFFFFF	Read only	-	O
0x0041 – 0x007F	Reserved					
0x0080	TokenCarrierID	Octet String	1 to 21	Read/ Write	-	O
0x0081 – 0x00FF	Reserved					

10427

10428 D.7.2.2.1.1 PaymentControl Configuration Attribute

10429 The *PaymentControlConfiguration* attribute represents the payment mechanisms currently
10430 enabled within the Metering Device. Bit encoding of this field is outlined in Table D-130.

10431 **Table D-130– *Payment Control Configuration* Attribute**

Bits	Description
0	Disconnection Enabled

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

1	Prepayment Enabled
2	Credit Management Enabled
3	Reserved
4	Credit Display Enabled
5	Reserved
6	Account Base
7	Contactor Fitted
8	Standing Charge Configuration
9	Emergency Standing Charge Configuration
10	Debt Configuration
11	Emergency Debt Configuration
12-15	Reserved

10432
10433 Examples for the setting of this attribute:

Mode of operation	**Description**	**Bits**										
		0	1	2	3	4	5	6	7	8	9	10
Credit Only	The meter is not fitted with a service interrupt device or the interrupt device is disabled. The meter does have an accounting function.	0	0	1	0	X	0	1	0	0	0	0
Credit with disconnect fitted	The meter is fitted with a service interrupt device which can be operated using the supply interrupt command.(for example, this mode allows the supply to the premises to be interrupted in the case of a change of tenancy).	1	0	1	0	X	0	1	1	0	0	0
Prepayment	The meter is fitted with a service interrupt device which can be operated using the supply interrupt command. The accounting function is enabled to allow the consumer's account balance to be shown in monetary values and when it reaches zero or a predefined limit, the supply will be interrupted by the meter. Additionally, the meter will respond to remote supply interruption commands	1	1	1	0	1	0	0	1	X	X	X

10434

10435 **Disconnection Enabled:** Indicates whether the metering device is to disconnect the energy
10436 supply on expiry of available credit.

10437 **Prepayment Enabled:** Indicates if the meter is a 'prepayment' meter; if this value is 0, the meter
10438 is considered to be a 'credit' meter.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Credit Management Enabled: Indicates whether the metering device should manage accounting functionality according to available tariff information.

Credit Display Enabled: Indicates whether the metering device should display the credit status.

Account Base: Indicates whether the metering device is running in Monetary (0) or Unit based (1) units. If Monetary based, the unit of measure is defined in the Price cluster, if Unit based, the unit of measure is defined in the Metering cluster

Contactor Fitted: Indicates whether the metering device is fitted with a Contactor i.e. is capable if disconnecting the energy supply.

Standing Charge Configuration: Indicates whether the standing charge collection is halted when the prepaid credit is exhausted.

Emergency Standing Charge Configuration: Indicates whether the standing charge collection is halted when the device is in Emergency Credit mode.

Debt Configuration: Indicates whether the debt collection is halted when the prepaid credit is exhausted.

Emergency Debt Configuration: Indicates whether the debt is collected when the device is in Emergency Credit mode.

D.7.2.2.1.2 Credit Remaining Attribute

The *Credit Remaining* attribute represents the amount of credit remaining on the Metering Device. If Monetary-based, this attribute is measured in a base unit of *Currency* with the decimal point located as indicated by the *Trailing Digits* field, as defined in the Price cluster. If Unit-based, the unit of measure is as defined in the Metering cluster (see sub-clause D.3.2.2.4.1).

D.7.2.2.1.3 Emergency Credit Remaining Attribute

The *Emergency Credit Remaining* attribute represents the amount of Emergency Credit still available on the Metering Device. If Monetary-based, this attribute is measured in a base unit of *Currency* with the decimal point located as indicated by the *Trailing Digits* field, as defined in the Price cluster. If Unit-based, the unit of measure is as defined in the Metering cluster (see sub-clause D.3.2.2.4.1).

D.7.2.2.1.4 Credit Status Attribute

The *Credit Status* attribute represents the current status of credit within the Metering Device. Bit encoding of this field is outlined in Table D-131. Explanation of the use of this attribute can be found in section D.7.4.1.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

10471 **Table D-131– *Credit Status* Attribute**

Bits	Description
0	Credit OK
1	Low Credit
2	Emergency Credit Enabled
3	Emergency Credit Available
4	Emergency Credit Selected
5	Emergency Credit In Use
6	Credit Exhausted
7	Reserved for Future Use

10472

10473 D.7.2.2.1.5 CreditRemainingTimeStamp Attribute

10474 The UTC time at which the *Credit Remaining* attribute was last populated.

10475 D.7.2.2.1.6 AccumulatedDebt Attribute

10476 The *AccumulatedDebt* attribute represents the total amount of debt remaining on the Metering
10477 Device. This attribute is always Monetary based and, as such, this attribute is measured in a base
10478 unit of *Currency* with the decimal point located as indicated by the *Trailing Digits* field, as
10479 defined in the Price cluster.

10480 D.7.2.2.1.7 OverallDebtCap Attribute

10481 The *OverallDebtCap* attribute represents the total amount of debt that can be taken from top-ups
10482 (in the case of multiple instantiated top-up based debts on the Metering Device). This attribute is
10483 configured to the required limit per unit time (fixed globally in the application at one week) that
10484 the consumer pays off against their debts. This attribute is always a monetary value, and as such
10485 this attribute is measured in a base unit of *Currency* with the decimal point located as indicated
10486 by the *Trailing Digits* field, as defined in the Price cluster.

10487 As an example, a consumer has a single Percentage Based debt in operation, with a collection
10488 rate of 20% and an *OverallDebtCap* of £5 per week. He buys £5 credit every day. Table D-132
10489 shows the resultant allocation of the amounts purchased:

10490 **Table D-132– OverallDebtCap Example**

	Amount Purchased	Amount to Debt	Amount to Credit
Monday	£5	20% = £1	£4
Tuesday	£5	20% = £1	£4
Wednesday	£5	20% = £1	£4
Thursday	£5	20% = £1	£4
Friday	£5	20% = £1	£4

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Saturday	£5	Cap reached	£5
Sunday	£5	Cap reached	£5

Once the cap value has been reached during a week then no further amounts are deducted from the purchases.

As an extension of the example, if the customer purchased £50 credit on the Monday, the meter would take £5 (not £10) and would also not take any further debt payments from any other purchases made in the same week.

D.7.2.2.1.8 EmergencyCreditLimit/Allowance Attribute

The *EmergencyCreditLimit/Allowance* attribute may be updated by the utility company. This is the amount of Emergency Credit available to loan to the consumer when the remaining balance goes below the low credit threshold. If Monetary based, then this attribute is measured in base unit of *Currency* with the decimal point located as indicated by the *Trailing Digits* field, as defined in the Price cluster. If Unit based, the unit of measure is as defined in the Metering cluster (see sub-clause D.3.2.2.4.1).

D.7.2.2.1.9 EmergencyCreditThreshold Attribute

When credit (or emergency credit) falls below this threshold, an alarm is raised to warn the consumer of imminent supply interruption and, if available, to offer Emergency Credit. If Monetary based, the unit of measure is the same as that defined in the Price cluster. If Unit based, the unit of measure is as defined in the Metering cluster (see sub-clause D.3.2.2.4.1).

D.7.2.2.1.10 TotalCreditAdded Attribute

An unsigned 48-bit integer value indicating running total of credit topped up to date. If Monetary based, this attribute is measured in a base unit of *Currency* with the decimal point located as indicated by the *Trailing Digits* field, as defined in the Price cluster. If Unit based, the unit of measure is as defined in the Metering cluster (see sub-clause D.3.2.2.4.1). At change of Tenant or Supplier, this attribute shall be reset to zero.

D.7.2.2.1.11 MaxCreditLimit Attribute

An unsigned 32-bit integer value indicating the maximum credit balance allowed on a meter. Any further top-up amount that will cause the meter to exceed this limit will be rejected. This value can be stated in currency (as per the Price cluster) or in units (unit of measure will be defined in the Metering cluster) depending on the Prepayment mode of operation defined in section D.7.2.2.1.1 (*Payment Control Configuration* attribute). A value of 0xFFFFFFFF shall indicate that this limit is disabled and that all further top-ups should be permitted.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

10522 D.7.2.2.1.12 MaxCreditPerTopUp Attribute

10523 An unsigned 32-bit integer value indicating the maximum credit per top-up. Any single top-up
10524 greater than this threshold will cause the meter to reject the top-up. This value can be stated in
10525 currency (as per the Price cluster) or in units (unit of measure will be defined in the Metering
10526 cluster) depending on the Prepayment mode of operation defined in section D.7.2.2.1.1 (*Payment*
10527 *Control Configuration* attribute). A value of 0xFFFFFFFF shall indicate that this parameter is
10528 disabled and that there should be no limit on the amount of allowed credit in a top-up.

10529 D.7.2.2.1.13 FriendlyCreditWarning Attribute

10530 An unsigned 8-bit integer value indicating the amount of time, in minutes, before the *Friendly*
10531 *Credit Period End Warning* alarm flag is triggered. The default value is 10 mins before the
10532 currently active Friendly Credit period is due to end.

10533 D.7.2.2.1.14 LowCreditWarningLevel Attribute

10534 An unsigned 32 bit integer that defines the **utility** low credit value below which the Low Credit
10535 warning should sound. The Low Credit warning shall be triggered when the value between the
10536 remaining credit and the disconnection point falls below this value. Falling below this value shall
10537 trigger the Low Credit warning alert within this cluster. The value is in a base unit of *Currency*
10538 (as per the Price cluster) or in Units (as per the Metering cluster). The attribute is set from the
10539 backhaul connection.

10540 D.7.2.2.1.15 IHDLowCreditWarningLevel Attribute

10541 An unsigned 32 bit integer that is defined by the **consumer** for a low credit value below which a
10542 Low Credit warning should sound. The Low Credit warning shall be triggered when the value
10543 between the remaining credit and the disconnection point falls below this value. This shall not
10544 trigger the Low Credit warning alert within this cluster. The value is in a base unit of *Currency*
10545 (as per the Price cluster) or in Units (as per the Metering cluster).

10546 D.7.2.2.1.16 InterruptSuspendTime Attribute

10547 When the end of a configured non-disconnect period is reached and the supply is to be
10548 interrupted due to insufficient credit being available, the meter will provide visual and audible
10549 alerts and the interruption will be suspended for a further period of minutes defined by this
10550 attribute. If no payments are applied to the meter during this period, or if insufficient credit is
10551 added, then, at the end of this period, an alert will be provided and the supply will then be
10552 interrupted.

10553 D.7.2.2.1.17 RemainingFriendlyCreditTime Attribute

10554 An unsigned 16-bit integer value indicating the amount of time remaining, in minutes, in a
10555 currently active Friendly Credit period. A value of zero shall indicate that no period is currently
10556 active (i.e. 0 = expired/no minutes left).

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

10557 D.7.2.2.1.18 NextFriendlyCreditPeriod Attribute

10558 The UTC time at which the next Friendly Credit period is due to commence.

10559 D.7.2.2.1.19 CutOffValue Attribute

10560 This attribute is a signed 32 bit integer that shall either be zero or a negative value (in all known
10561 cases). This allowance is measured in base unit of *Currency* with the decimal point located as
10562 indicated by the *Trailing Digits* field, as defined in the Price cluster. If Unit based, the unit of
10563 measure is as defined in the Metering cluster (see sub-clause D.3.2.2.4.1).

10564 This attribute represents a threshold relating to the absolute value of the *CreditRemaining*
10565 attribute, that when reached (when credit is decrementing) causes the supply of service to be
10566 disconnected. There can be several types of credit within a payment metering system of which
10567 there are 2 specified in this specification (*Credit* and *EmergencyCredit*). The *CreditRemaining*
10568 attribute shall contain the net worth of a consumers account within the meter, consolidating all
10569 active credit types (both *Credit* and *EmergencyCredit* if in use). As *EmergencyCredit* is
10570 effectively a loan from the supplier it becomes a liability once it is used, and when it is exhausted
10571 will force the *RemainingCredit* to a negative value. There are a number of other factors that can
10572 affect the way a prepayment meter works and which values are displayed to the end consumer.
10573 However, when a meter's *EmergencyCredit* has run out, the *CreditRemaining* value shall contain
10574 the total liability of the consumer (that he is required to pay before *EmergencyCredit* shall be
10575 available again) as a negative value.

10576 D.7.2.2.1.20 TokenCarrierId Attribute

10577 The *TokenCarrierId* attribute provides a method for utilities to publish the payment card number
10578 that is used with this meter set. The *TokenCarrierId* attribute is a ZCL Octet String capable of
10579 storing a 20 character string (the first Octet indicates length) encoded in the UTF-8 format. The
10580 *TokenCarrierId* attribute represents the current active value for the property.

10581 D.7.2.2.2 **Top-up Attribute Set**

10582 The following set of attributes provides access to previous successful credit *top-ups* on a
10583 prepayment meter. #1 is the most recent, based on time.

10584 **Table D-133– Top-up Attribute Set**

Identifier	Name	Type	Range	Access	Default	Mandatory /Optional
0x0100	*Top up Date/Time #1*	UTCTime		Read only	-	O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x0101	*Top up Amount #1*	Signed 32-bit Integer	-0x7FFFFFFF to +0x7FFFFFFF	Read only	-	O
0x0102	*Originating Device #1*	8 bits Enumeration	0x00 to 0xFF	Read only	-	O
0x0103	*Top up Code #1*	Octet String	1-26	Read only	-	O
0x0104-0x010F	Reserved					
0x0110	*Top up Date/Time #2*	UTCTime		Read only	-	O
0x0111	*Top up Amount #2*	Signed 32-bit Integer	-0x7FFFFFFF to +0x7FFFFFFF	Read only	-	O
0x0112	*Originating Device #2*	8 bits Enumeration	0x00 to 0xFF	Read only	-	O
0x0113	*Top up Code #2*	Octet String	1-26	Read only	-	O
0x0114-0x011F	Reserved					
0x0120	*Top up Date/Time #3*	UTCTime		Read only	-	O
0x0121	*Top up Amount #3*	Signed 32-bit Integer	-0x7FFFFFFF to +0x7FFFFFFF	Read only	-	O
0x0122	*Originating Device #3*	8 bits Enumeration	0x00 to 0xFF	Read only	-	O
0x0123	*Top up Code #3*	Octet String	1-26	Read only	-	O
0x0124-0x012F	Reserved					
0x0130	*Top up Date/Time #4*	UTCTime		Read only	-	O
0x0131	*Top up Amount #4*	Signed 32-bit Integer	-0x7FFFFFFFFFFF to +0x7FFFFFFFFFFF	Read only	-	O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x0132	*Originating Device #4*	8 bits Enumeration	0x00 to 0xFF	Read only	-	O
0x0133	*Top up Code #4*	Octet String	1-26	Read only	-	O
0x0134-0x013F	Reserved					
0x0140	*Top up Date/ Time#5*	UTCTime		Read only	-	O
0x0141	*Top up Amount #5*	Signed 32-bit Integer	-0x7FFFFFFF to +0x7FFFFFFF	Read only	-	O
0x0142	*Originating Device #5*	8 bits Enumeration	0x00 to 0xFF	Read only	-	O
0x0143	*Top up Code #5*	Octet String	1-26	Read only	-	O
0x0144-0x01FF	Reserved					

10585

10586 D.7.2.2.2.1 Top up Date/Time Attribute

10587 The *Top up Date/Time* attribute represents the time that the credit was topped up on the
10588 Metering Device. There are five records containing this attribute, one for each of the last five
10589 top-ups.

10590 D.7.2.2.2.2 Top up Amount Attribute

10591 The *Top up Amount* attribute represents the amount of credit that was added to the Metering
10592 Device during the top up. If Monetary-based, this attribute is measured in a base unit of
10593 *Currency* with the decimal point located as indicated by the *Trailing Digits* field, as defined
10594 in the Price cluster. If Unit-based, the unit of measure is as defined in the Metering cluster
10595 (see sub-clause D.3.2.2.4.1). There are five records containing this attribute, one for each of the
10596 last five top-ups.

10597 D.7.2.2.2.3 Originating Device Attribute

10598 The *Originating Device* attribute represents the SE device that was the source of the top-up
10599 command. The enumerated values of this field are outlined in Table D-146. There are five
10600 records containing this attribute, one for each of the last five top-ups.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

10601 D.7.2.2.2.4 Top up Code Attribute

10602 The *Top up Code* attribute represents any encrypted number that was used to apply the credit to
10603 the meter; the octet string shall be as it was received, i.e. not decoded. There are five records
10604 containing this attribute, one for each of the last five top-ups.

10605 D.7.2.2.3 **Debt Attribute Set**

10606 The following set of attributes provides access to information on debt held on a Prepayment
10607 meter.

10608 **Table D-134– Debt Attribute Set**

Identifier	Name	Type	Range	Access	Default	Mandatory /Optional
0x0200 – 0x020F	Reserved					
0x0210	DebtLabel#1	Octet string	1-13	Read only	-	O
0x0211	DebtAmount#1	Unsigned 32-bit Integer	0x00000000 – 0xFFFFFFFF	Read only	-	O
0x0212	DebtRecovery Method#1	8 bits Enumeration	0x00 – 0xFF	Read only	-	O
0x0213	DebtRecovery StartTime#1	UTCTime		Read only	-	O
0x0214	DebtRecovery CollectionTime#1	Unsigned 16 - bit integer	0x0000 – 0x05A0	Read only	0	O
0x0215	Reserved					
0x0216	DebtRecovery Frequency#1	8 bits Enumeration	0x00 – 0xFF	Read only	-	O
0x0217	DebtRecovery Amount#1	Unsigned 32-bit Integer	0x00000000 – 0xFFFFFFFF	Read only	-	O
0x0218	Reserved					
0x0219	DebtRecovery TopUpPercentage#1	Unsigned 16 - bit integer	0x0000 – 0x2710	Read only	0	O
0x021A – 0x021F	Reserved					
0x0220	DebtLabel#2	Octet string	1-13	Read only	-	O
0x0221	DebtAmount#2	Unsigned 32-bit Integer	0x00000000 – 0xFFFFFFFF	Read only	-	O
0x0222	DebtRecovery Method#2	8 bits Enumeration	0x00 – 0xFF	Read only	-	O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x0223	DebtRecovery StartTime#2	UTCTime		Read only	-	O
0x0224	DebtRecovery CollectionTime#2	Unsigned 16 - bit integer	0x0000 – 0x05A0	Read only	0	O
0x0225	Reserved					
0x0226	DebtRecovery Frequency#2	8 bits Enumeration	0x00 – 0xFF	Read only	-	O
0x0227	DebtRecovery Amount#2	Unsigned 32-bit Integer	0x00000000 – 0xFFFFFFFF	Read only	-	O
0x0228	Reserved					
0x0229	DebtRecovery TopUpPercentage#2	Unsigned 16 - bit integer	0x0000 – 0x2710	Read only	0	O
0x022 A – 0x022F	Reserved					
0x0230	DebtLabel#3	Octet string	1-13	Read only	-	O
0x0231	DebtAmount#3	Unsigned 32-bit Integer	0x00000000 – 0xFFFFFFFF	Read only	-	O
0x0232	DebtRecovery Method#3	8 bits Enumeration	0x00 – 0xFF	Read only	-	O
0x0233	DebtRecovery StartTime#3	UTCTime		Read only	-	O
0x0234	DebtRecovery CollectionTime#3	Unsigned 16 - bit integer	0x0000 – 0x05A0	Read only	0	O
0x0235	Reserved					
0x0236	DebtRecovery Frequency#3	8 bits Enumeration	0x00 – 0xFF	Read only	-	O
0x0237	DebtRecovery Amount#3	Unsigned 32-bit Integer	0x00000000 – 0xFFFFFFFF	Read only	-	O
0x0238	Reserved					
0x0239	DebtRecovery TopUpPercentage#3	Unsigned 16 - bit integer	0x0000 – 0x2710	Read only	0	O
0x023 A – 0x02FF	Reserved					

10609

10610 D.7.2.2.3.1 DebtLabel#N Attribute

10611 The *DebtLabel#n* attribute provides a method for utilities to assign a name to a particular type of
10612 debt. The *DebtLabel#n* attribute is a ZCL Octet String field capable of storing a 12 character

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

10613 string (the first Octet indicates length) encoded in the UTF-8 format. This applies to all debt
10614 recovery methods.

10615 D.7.2.2.3.2 DebtAmount#N Attribute

10616 An unsigned 32-bit field to denote the amount of Debt remaining on the Metering Device. This
10617 parameter shall be measured in base unit of *Currency* with the decimal point located as indicated
10618 by the *Trailing Digits* field, as defined in the Price Cluster.

10619 D.7.2.2.3.3 DebtRecoveryMethod#N Attribute

10620 An enumerated attribute denoting the debt recovery method used for this debt type. The
10621 enumerated values for this field are outlined in Table D-135 (Time based, Percentage based and
10622 Catch-Up based). This applies to all debt recovery methods.

10623 **Table D-135– Debt Recovery Method Enumerations**

Enumerated Value	Recovery Method
0x00	Time Based
0x01	Percentage Based
0x02	Catch-Up Based (Fixed Period)
0x03 – 0xFF	Reserved

10624

10625 D.7.2.2.3.4 DebtRecoveryStartTime#N Attribute

10626 A UTC Time field to denote the time at which the debt collection should start. This applies to all
10627 debt recovery methods.

10628 D.7.2.2.3.5 DebtRecoveryCollectionTime#N Attribute

10629 An unsigned 16-bit field denoting the time of day when the debt collection takes place. It is
10630 encoded as the number of minutes after midnight and has a valid range 0 .. 1440 with a default
10631 value of 0. This applies to all debt recovery methods.

10632 D.7.2.2.3.6 DebtRecoveryFrequency#N Attribute

10633 The *DebtRecoveryFrequency#N* attribute represents the period over which each
10634 *DebtRecoveryAmount#N* is recovered. The enumerated values of this field are outlined in Table
10635 D-136.

10636 **Table D-136–*Recovery Frequency* Field Enumerations**

Enumerated Value	Recovery Period
0x00	Per Hour
0x01	Per Day
0x02	Per Week

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x03	Per Month
0x04	Per Quarter

10637

10638 D.7.2.2.3.7 DebtRecoveryAmount#N Attribute

10639 The *DebtRecoveryAmount#N* attribute represents the amount of Debt recovered each period
10640 specified by *DebtRecoveryFrequency#N*, measured in base unit of *Currency* with the decimal
10641 point located as indicated by the *Trailing Digits* field, as defined in the Price Cluster. This
10642 attribute only applies to Time based and Catch-Up based debt recovery. A value of 0 indicates
10643 not used.

10644

10645 D.7.2.2.3.8 DebtRecoveryTopUpPercentage#N Attribute

10646 An unsigned 16-bit field used in Percentage based recovery to denote the percentage from a top-
10647 up amount to be deducted from the debt. For example, if the *DebtRecoveryTopUpPercentage#N*
10648 is set to 10% and the customer topped up the device with 10 units of Currency, then 1 unit is
10649 deducted from the amount being topped up and paid towards the debt recovery, i.e the device is
10650 credited with only 9 units of currency. The percentage is always in the following format xxx.xx.
10651 The default is 0.00% and maximum value is 100.00%.

10652 D.7.2.2.4 **Supply Control Set**

10653 The Supply Control functionality has been moved to the Metering cluster (see Annex D.3 for
10654 further details).

10655

10656 D.7.2.2.5 **Alarms Attribute Set**

10657 The following set of attributes provides a means to control which prepayment alarms may be
10658 generated from the meter.

10659 **Table D-137– Alarm Attribute Set**

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0400	PrepaymentAlarmStatus	16 bit BitMap	0x0000 to 0xFFFF	Read/Write	0xFFFF	O
0x0401	PrepayGenericAlarmMask	16-bit BitMap	0x0000 to 0xFFFF	Read/Write	0xFFFF	O
0x0402	PrepaySwitchAlarmMask	16-bit BitMap	0x0000 to 0xFFFFF	Read/Write	0xFFFF	O
0x0403	PrepayEventAlarmMask	16-bit BitMap	0x0000 to 0xFFFF	Read/Write	0xFFFF	O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

10660

10661 D.7.2.2.5.1 Prepayment Alarm Status Attribute

10662 The *PrepaymentAlarmStatus* attribute provides indicators reflecting the current error conditions
10663 found by the prepayment metering device. This attribute is a 16-bit field where when an
10664 individual bit is set, an error or warning condition exists. The behaviour causing the setting or
10665 resetting of each bit is device specific. In other words, the application within the prepayment
10666 metering device will determine and control when these settings are either set or cleared. The ESI
10667 should make alarms available to upstream systems, together with consumption data collected
10668 from a battery operated meter.

10669 **Table D-138– Prepayment Alarm Status Indicators**

Bit field	Alarm Condition	Meaning / Description
0	Low Credit Warning	An alarm triggered by a configured threshold.
1	Top Up Code Error	The Top up code has been sent but it is too long or short for the meter
2	Top Up Code Already Used	The Top up code has been sent but the credit value for this top up code has already been applied and this is a duplicate request.
3	Top Up Code Invalid	The Top up code is a correct length but is not a valid top up code.
4	Friendly Credit In Use	The meter is in a Friendly Credit period and Friendly Credit is being used due to no actual credit being available on the meter.
5	Friendly Credit Period End Warning	This is triggered when the time remaining in a Friendly Credit period falls below the value of the FriendlyCreditWarning attribute (default 1hr) and the above Friendly Credit In Use flag is set.
6	EC Available	An alarm triggered when Emergency credit is available to be selected
7	Unauthorised Energy Use	GAS: Valve Fault and unauthorised gas is being provided to the home ELECTRICITY: Disconnection Fault and unauthorised electricity is being provided to the house.
8	Disconnected Supply Due to Credit	Supply has been disconnected due to no credit on meter. Cleared by addition of credit or by selecting Emergency Credit
9	Disconnected Supply Due to Tamper	Supply has been disconnected due to a tamper detect on the meter. It can also be due to a fault on the meter that is not covered by another flag.
10	Disconnected Supply Due to HES	This is normally due to the HES cutting the supply
11	Physical Attack	Physical attack on the Prepayment Meter
12	Electronic Attack	Electronic attack on the Prepayment Meter
13	Manufacture Alarm Code A	Manufacture Alarm Code A
14	Manufacture Alarm Code B	Manufacture Alarm Code B
15	Reserved	

10670

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

10671 D.7.2.2.5.2 Alarm Mask Attributes

10672 The Alarm Mask attributes of the Alarms Attribute Set specify whether each of the alarms listed
10673 in the corresponding alarm group in Table D-139 through Table D-142 is enabled. When the bit
10674 number corresponding to the alarm number (minus the group offset) is set to 1, the alarm is
10675 enabled, else it is disabled. Bits not corresponding to a code in the respective table are reserved.

10676 D.7.2.2.5.3 Alarm Codes

10677 The alarm codes are organised in logical groups corresponding to the types of activity as listed
10678 below. The three main alarm groups are: GenericAlarmMask, PrepaySwitchAlarmMask, and
10679 PrepayEventAlarmMask.

10680 **Table D-139– Alarms Code Group**

Enumerated Alarm Codes	Alarm Condition
0x00 – 0x0F	PrePayGenericAlarmGroup
0x10 – 0x1F	PrepaySwitchAlarmGroup
0x20 – 0x4F	PrepayEventAlarmGroup
0x50 – 0xFF	Reserved

10681

10682 The Alarms that can be enabled/disabled in the PrepayGenericAlarmGroup are as follows:

10683 **Table D-140– PrepayGenericAlarmGroup**

Enumerated Alarm Code	Alarm Condition
0x00	Low Credit (for all types of credit)
0x01	No Credit (Zero Credit)
0x02	Credit Exhausted
0x03	Emergency Credit Enabled
0x04	Emergency Credit Exhausted
0x05	IHD Low Credit Warning
0x06	Event Log Cleared
0x07 - 0x0F	Reserved

10684

10685 The Alarms that can be enabled/disabled in the *PrepaySwitchAlarmGroup* are as follows:

10686 **Table D-141– PrepaySwitchAlarmGroup**

Enumerated Alarm Code	Alarm Condition
0x10	Supply ON
0x11	Supply ARM
0x12	Supply OFF
0x13	Disconnection Failure (Shut Off Mechanism Fail)

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Enumerated Alarm Code	Alarm Condition
0x14	Disconnected due to Tamper Detected.
0x15	Disconnected due to Cut off Value.
0x16	Remote Disconnected.
0x17 – 0x1F	Reserved

10687

10688 The Alarms that can be enabled/disabled in the *PrepayEventAlarmGroup* are as follows:

10689 **Table D-142– PrepayEventAlarmGroup**

Enumerated Alarm Code	Alarm Condition
0x20	Physical Attack on the Prepay Meter
0x21	Electronic Attack on the Prepay Meter
0x22	Discount Applied
0x23	Credit Adjustment
0x24	Credit Adjustment Fail
0x25	Debt Adjustment
0x26	Debt Adjustment Fail
0x27	Mode Change
0x28	Topup Code Error
0x29	Topup Already Used
0x2A	Topup Code Invalid
0x2B	Friendly Credit In Use
0x2C	Friendly Credit Period End Warning
0x2D	Friendly Credit Period End
0x2E-0x2F	Reserved
0x30	ErrorRegClear
0x31	AlarmRegClear
0x32	Prepay Cluster Not Found
0x33-0x40	Reserved
0x41	ModeCredit2Prepay
0x42	ModePrepay2Credit
0x43	ModeDefault
0x44-0x4F	Reserved

10690

10691 D.7.2.2.6 **Historical Cost Consumption Information Set**

10692 **Table D-143– Historical Cost Consumption Information Attribute Set**

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x0500	HistoricalCostConsumption Formatting	8-bit BitMap	0x00 to 0xFF	Read Only	-	O
0x0501	Consumption UnitofMeasurement	8-bit Enumeration	0x00 to 0xFF	Read Only	0x00	O
0x0502	CurrencyScalingFactor	8-bit Enumeration	0x00 to 0xFF	Read Only	-	O
0x0503	Currency	Unsigned 16-bit Integer	0x0000 to 0x0FFF	Read Only	-	O
0x0504-0x051B	Reserved					
0x051C	CurrentDay CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O
0x051D	CurrentDay CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O
0x051E	PreviousDay CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O
0x051F	PreviousDay CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O
0x0520	PreviousDay2 CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O
0x0521	PreviousDay2 CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O
0x0522	PreviousDay3 CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O
0x0523	PreviousDay3 CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O
0x0524	PreviousDay4 CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O
0x0525	PreviousDay4 CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x0526	PreviousDay5 CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O
0x0527	PreviousDay5 CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O
0x0528	PreviousDay6 CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O
0x0529	PreviousDay6 CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O
0x052A	PreviousDay7 CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O
0x052B	PreviousDay7 CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O
0x052C	PreviousDay8 CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O
0x052D	PreviousDay8 CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O
0x052E-0x052F	Reserved					
0x0530	CurrentWeek CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O
0x0531	CurrentWeek CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O
0x0532	PreviousWeek CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O
0x0533	PreviousWeek CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O
0x0534	PreviousWeek2 CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O
0x0535	PreviousWeek2 CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x0536	PreviousWeek3 CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O
0x0537	PreviousWeek3 CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O
0x0538	PreviousWeek4 CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O
0x0539	PreviousWeek4 CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O
0x053A	PreviousWeek5 CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O
0x053B	PreviousWeek5 CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O
0x053C-0x053F	Reserved					
0x0540	CurrentMonth CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O
0x0541	CurrentMonth CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O
0x0542	PreviousMonth CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O
0x0543	PreviousMonth CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O
0x0544	PreviousMonth2 CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O
0x0545	PreviousMonth2 CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O
0x0546	PreviousMonth3 CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x0547	PreviousMonth3 CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O
0x0548	PreviousMonth4 CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O
0x0549	PreviousMonth4 CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O
0x054A	PreviousMonth5 CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O
0x054B	PreviousMonth5 CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O
0x054C	PreviousMonth6 CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O
0x054D	PreviousMonth6 CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O
0x054E	PreviousMonth7 CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O
0x054F	PreviousMonth7 CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O
0x0550	PreviousMonth8 CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O
0x0551	PreviousMonth8 CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O
0x0552	PreviousMonth9 CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O
0x0553	PreviousMonth9 CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O
0x0554	PreviousMonth10 CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x0555	PreviousMonth10 CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O
0x0556	PreviousMonth11 CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O
0x0557	PreviousMonth11 CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O
0x0558	PreviousMonth12 CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O
0x0559	PreviousMonth12 CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O
0x055A	PreviousMonth13 CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O
0x055B	PreviousMonth13 CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFF	Read Only	-	O
0x055C	Historical Freeze Time	Unsigned 16-bit Integer	0x0000 to 0x173C	Read Only	0x0000	O
0x055D-0x05FF	Reserved					

10693

10694 D.7.2.2.6.1 HistoricalCostConsumptionFormatting Attribute

10695 *HistoricalCostConsumptionFormatting* provides a method to properly decipher the decimal point
10696 location for the values found in the Historical Cost Consumption Set of attributes. The most
10697 significant nibble indicates the number of digits to the left of the decimal point, the least
10698 significant nibble the number of digits to the right.

10699 This attribute shall be used against the following attributes:

10700 • *CurrentDayCostConsumptionDelivered*
10701 • *CurrentDayCostConsumptionReceived*
10702 • *PreviousDayNCostConsumptionDelivered*
10703 • *PreviousDayNCostConsumptionReceived*
10704 • *CurrentWeekCostConsumptionDelivered*
10705 • *CurrentWeekCostConsumptionReceived*

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

- *PreviousWeekNCostConsumptionDelivered*
- *PreviousWeekNCostConsumptionReceived*
- *CurrentMonthCostConsumptionDelivered*
- *CurrentMonthCostConsumptionReceived*
- *PreviousMonthNCostConsumptionDelivered*
- *PreviousMonthNCostConsumptionReceived*

D.7.2.2.6.2 ConsumptionUnitofMeasurement Attribute

ConsumptionUnitofMeasurement provides a label for the Energy, Gas, or Water being measured by the metering device. This attribute is an 8-bit enumerated field. The bit descriptions for this attribute are listed in Table D-26.

This attribute shall be used against the following attributes:

- *CurrentDayCostConsumptionDelivered*
- *CurrentDayCostConsumptionReceived*
- *PreviousDayNCostConsumptionDelivered*
- *PreviousDayNCostConsumptionReceived*
- *CurrentWeekCostConsumptionDelivered*
- *CurrentWeekCostConsumptionReceived*
- *PreviousWeekNCostConsumptionDelivered*
- *PreviousWeekNCostConsumptionReceived*
- *CurrentMonthCostConsumptionDelivered*
- *CurrentMonthCostConsumptionReceived*
- *PreviousMonthNCostConsumptionDelivered*
- *PreviousMonthNCostConsumptionReceived*

D.7.2.2.6.3 CurrencyScalingFactor Attribute

C*urrencyScalingFactor* provides a scaling factor for the *Currency* attribute for the Energy, Gas, or Water being measured by the metering device. This attribute is an 8-bit enumeration, the enumerated values for which are outlined in Table D-144. Note that this attribute will allow for a different resolution for historical values compared to values in the Price cluster.

This attribute shall be used against the following attributes:

- *CurrentDayCostConsumptionDelivered*
- *CurrentDayCostConsumptionReceived*
- *PreviousDayNCostConsumptionDelivered*
- *PreviousDayNCostConsumptionReceived*

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

- *CurrentWeekCostConsumptionDelivered*
- *CurrentWeekCostConsumptionReceived*
- *PreviousWeekNCostConsumptionDelivered*
- *PreviousWeekNCostConsumptionReceived*
- *CurrentMonthCostConsumptionDelivered*
- *CurrentMonthCostConsumptionReceived*
- *PreviousMonthNCostConsumptionDelivered*
- *PreviousMonthNCostConsumptionReceived*

Table D-144–*CurrencyScalingFactor* Enumerations

Enumerated Value	Scaling Factor
0x00	x 10^{-6}
0x01	x 10^{-5}
0x02	x 10^{-4}
0x03	x 10^{-3}
0x04	x 10^{-2}
0x05	x 10^{-1}
0x06	x 1
0x07	x 10
0x08	x 100
0x09	x 10^{3}
0x0A	x 10^{4}
0x0B	x 10^{5}
0x0C	x 10^{6}

D.7.2.2.6.4 Currency Attribute

The *Currency* attribute provides the currency for the Energy, Gas, or Water being measured by the prepayment device. The value of the attribute should match one of the values defined by ISO 4217. This unsigned 16-bit value indicates the currency in which the following attributes are represented:

- *CurrentDayCostConsumptionDelivered*
- *CurrentDayCostConsumptionReceived*
- *PreviousDayNCostConsumptionDelivered*
- *PreviousDayNCostConsumptionReceived*
- *CurrentWeekCostConsumptionDelivered*

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

- *CurrentWeekCostConsumptionReceived*
- *PreviousWeekNCostConsumptionDelivered*
- *PreviousWeekNCostConsumptionReceived*
- *CurrentMonthCostConsumptionDelivered*
- *CurrentMonthCostConsumptionReceived*
- *PreviousMonthNCostConsumptionDelivered*
- *PreviousMonthNCostConsumptionReceived*

D.7.2.2.6.5 CurrentDayCostConsumptionDelivered Attribute

CurrentDayCostConsumptionDelivered represents the summed value of Energy, Gas, or Water delivered to the premises since the HFT. If optionally provided, *CurrentDayCostConsumptionDelivered* is updated continuously as new measurements are made. If the optional *Historical Freeze Time* attribute is not available, default to midnight local time.

D.7.2.2.6.6 CurrentDayCostConsumptionReceived Attribute

CurrentDayCostConsumptionReceived represents the summed value of Energy, Gas, or Water received from the premises since the HFT. If optionally provided, *CurrentDayCostConsumptionReceived* is updated continuously as new measurements are made. If the optional *Historical Freeze Time* attribute is not available, default to midnight local time.

D.7.2.2.6.7 PreviousDayNCostConsumptionDelivered Attribute

PreviousDayNCostConsumptionDelivered represents the summed value of Energy, Gas, or Water delivered to the premises within the previous 24 hour period starting at the HFT. If the optional *Historical Freeze Time* attribute is not available, default to midnight local time.

D.7.2.2.6.8 PreviousDayNCostConsumptionReceived Attribute

PreviousDayNCostConsumptionReceived represents the summed value of Energy, Gas, or Water received from the premises within the previous 24 hour period starting at the HFT. If the optional *Historical Freeze Time* attribute is not available, default to midnight local time.

D.7.2.2.6.9 CurrentWeekCostConsumptionDelivered Attribute

CurrentWeekCostConsumptionDelivered represents the summed value of Energy, Gas, or Water delivered to the premises since the HFT on Monday to the last HFT read. If optionally provided, *CurrentWeekCostConsumptionDelivered* is updated continuously as new measurements are made. If the optional *Historical Freeze Time* attribute is not available, default to midnight local time.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

10795 D.7.2.2.6.10 CurrentWeekCostConsumptionReceived Attribute

10796 *CurrentWeekCostConsumptionReceived* represents the summed value of Energy, Gas, or Water
10797 received from the premises since the HFT on Monday to the last HFT read. If optionally
10798 provided, *CurrentWeekCostConsumptionReceived* is updated continuously as new measurements
10799 are made. If the optional *Historical Freeze Time* attribute is not available, default to midnight
10800 local time.

10801 D.7.2.2.6.11 PreviousWeekNCostConsumptionDelivered Attribute

10802 *PreviousWeekNCostConsumptionDelivered* represents the summed value of Energy, Gas, or
10803 Water delivered to the premises within the previous week period starting at the HFT on the
10804 Monday to the Sunday. If the optional *Historical Freeze Time* attribute is not available, default to
10805 midnight local time.

10806 D.7.2.2.6.12 PreviousWeekNCostConsumptionReceived Attribute

10807 *PreviousWeekNCostConsumptionReceived* represents the summed value of Energy, Gas, or
10808 Water received from the premises within the previous week period starting at the HFT on the
10809 Monday to the Sunday. If the optional *Historical Freeze Time* attribute is not available, default to
10810 midnight local time.

10811 D.7.2.2.6.13 CurrentMonthCostConsumptionDelivered Attribute

10812 *CurrentMonthCostConsumptionDelivered* represents the summed value of Energy, Gas, or Water
10813 delivered to the premises since the HFT on the 1st of the month to the last HFT read. If optionally
10814 provided, *CurrentMonthCostConsumptionDelivered* is updated continuously as new
10815 measurements are made. If the optional *Historical Freeze Time* attribute is not available, default
10816 to midnight local time.

10817 D.7.2.2.6.14 CurrentMonthCostConsumptionReceived Attribute

10818 *CurrentMonthCostConsumptionReceived* represents the summed value of Energy, Gas, or Water
10819 received from the premises since the HFT on the 1st of the month to the last HFT read. If
10820 optionally provided, *CurrentMonthCostConsumptionReceived* is updated continuously as new
10821 measurements are made. If the optional *Historical Freeze Time* attribute is not available, default
10822 to midnight local time.

10823 D.7.2.2.6.15 PreviousMonthNCostConsumptionDelivered Attribute

10824 *PreviousMonthNCostConsumptionDelivered* represents the summed value of Energy, Gas, or
10825 Water delivered to the premises within the previous Month period starting at the HFT on the 1st
10826 of the month to the last day of the month. If the optional *Historical Freeze Time* attribute is not
10827 available, default to midnight local time.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

10828 D.7.2.2.6.16 PreviousMonthNCostConsumptionReceived Attribute

10829 *PreviousMonthNCostConsumptionReceived* represents the summed value of Energy, Gas, or
10830 Water received from the premises within the previous month period starting at the HFT on the 1st
10831 of the month to the last day of the month. If the optional *Historical Freeze Time* attribute is not
10832 available, default to midnight local time.

10833 D.7.2.2.6.17 HistoricalFreezeTime Attribute

10834 *HistoricalFreezeTime* represents the time of day, in Local Time, when Historical Cost
10835 Consumption attributes are captured. *HistoricalFreezeTime* is an unsigned 16-bit value
10836 representing the hour and minutes for HFT. The byte usages are:

10837 **Bits 0 to 7:** Range of 0 to 0x3B representing the number of minutes past the top of the hour.
10838
10839 **Bits 8 to 15:** Range of 0 to 0x17 representing the hour of the day (in 24-hour format).

10840

10841 ### D.7.2.3 Commands Received

10842 Table D-145 lists cluster-specific commands that are received by the server.

10843

10844 **Table D-145– Cluster -specific Commands Received by the Server**

Command Identifier FieldValue	Description	Mandatory/ Optional
0x00	*Select Available Emergency Credit*	O
0x01	Reserved	
0x02	*Change Debt*	O
0x03	*Emergency Credit Setup*	O
0x04	*Consumer Top Up*	O
0x05	*CreditAdjustment*	O
0x06	*Change Payment Mode*	O
0x07	*Get Prepay Snapshot*	O
0x08	*Get Top Up Log*	O
0x09	*Set Low Credit Warning Level*	O
0x0A	*Get Debt Repayment Log*	O
0x0B	*Set Maximum Credit Limit*	O
0x0C	*Set Overall Debt Cap*	O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

10845

10846 D.7.2.3.1 **Select Available Emergency Credit Command**

10847 This command is sent to the Metering Device to activate the use of any Emergency
10848 Credit available on the Metering Device.

10849 D.7.2.3.1.1 Payload Format

Octets	4	1
Data Type	UTCTime	8 bits Enumeration
Field Name	Command Issue Date/ Time (M)	Originating Device (M)

10850 **Figure D-115– *Select Available Emergency Credit***
10851 **Command Payload**

10852 D.7.2.3.1.2 Payload Details

10853 **Command Issue Date/Time (mandatory):** A UTCTime field to indicate the date and time at
10854 which the selection command was issued.

10855 **Originating Device (mandatory):** An 8-bit enumeration field identifying the SE device issuing
10856 the selection command, using the lower byte of the Device ID defined in Table 5-14,
10857 and summarized in Table D-146.

10858 **Table D-146– Originating Device Field Enumerations**

Enumerated Value	Device
0x00	Energy Service Interface
0x01	Meter
0x02	In-Home Display Device
0x03 – 0xFF	Reserved

10859

10860

10861 D.7.2.3.1.3 Effect on Receipt

10862 A Mirroring device receiving this command shall return a ZCL Default Response with a status
10863 code of NOTIFICATION_PENDING. If the command buffer on the mirror is already full, the
10864 mirror shall instead return a ZCL Default Response to the initiating device with a status code of
10865 INSUFFICIENT_SPACE. The Mirroring device may timeout the buffered message, in which
10866 case it shall return a ZCL Default Response with a status code of TIMEOUT (see D.3.4.4.3 for
10867 further details).

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

10868

10869 D.7.2.3.2 **Change Supply Command**

10870 The *Change Supply* command has been moved to the Metering cluster (see Annex D.3 for further
10871 details).

10872 D.7.2.3.3 **Change Debt Command**

10873 The *ChangeDebt* command is sent to the Metering Device to change the debt values.

10874 D.7.2.3.3.1 Payload Format

Octets	**4**	**1-13**	**4**	**1**	**1**	**4**	**2**
Data Type	Unsigned 32-bit Integer	Octet String	Signed 32-bit Integer	8-bit Enumeration	8-bit Enumeration	UTCTime	Unsigned 16-bit Integer
Field Name	Issuer Event ID (M)	Debt Label (M)	Debt Amount (M)	Debt Recovery Method (M)	Debt Amount Type (M)	Debt Recovery Start Time (M)	Debt Recovery Collection Time (M)

10875

1	**4**	**2**
8-bit Enumeration	Signed 32-bit Integer	Unsigned 16-bit Integer
Debt Recovery Frequency (M)	Debt Recovery Amount (M)	Debt Recovery Balance Percentage (M)

10876 **Figure D-116– Change Debt Command Payload**

10877 D.7.2.3.3.2 Payload Details

10878 **Issuer Event Id (mandatory):** Unique identifier generated by the commodity provider. When
10879 new information is provided that replaces older information for the same time period, this field
10880 allows devices to determine which information is newer. The value contained in this field is a
10881 unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)
10882 identifying when the command was issued. Thus, newer information will have a value in the
10883 *Issuer Event ID* field that is larger than older information.

10884 **DebtLabel (mandatory):** The format and use of this field is the same as for the *DebtLabel#N*
10885 attribute as defined in D.7.2.2.3.1. A value of 0xFF in the first Octet (length) shall indicate that
10886 the value of this parameter shall remain unchanged on the Metering device following receipt of
10887 this command.

10888 **DebtAmount (mandatory):** The format and use of this field is the same as for the
10889 *DebtAmount#N* attribute as defined in D.7.2.2.3.2. A *DebtAmount* of 0xFFFFFFFF shall indicate
10890 that the value of this parameter shall remain unchanged on the Metering device following receipt
10891 of this command.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

DebtRecoveryMethod (mandatory): The format and use of this field is the same as for the *DebtRecoveryMethod#N* attribute as defined in D.7.2.2.3.3. A *DebtRecoveryMethod* of 0xFF shall indicate that the value of this parameter shall remain unchanged on the Metering device following receipt of this command.

DebtAmountType (mandatory): An 8-bit enumeration field identifying the type of debt information to be issued within this command. The Types are detailed in Table D-147 below:

Table D-147– Debt Amount Type Field Enumerations

Enumerated Value	Debt Type
0x00	Type 1 Absolute
0x01	Type 1 Incremental
0x02	Type 2 Absolute
0x03	Type 2 Incremental
0x04	Type 3 Absolute
0x05	Type 3 Incremental
0x06 – 0xFF	Reserved

DebtRecoveryStartTime (mandatory): The format and use of this field is the same as for the *DebtRecoveryStartTime#N* attribute as defined in D.7.2.2.3.4. A *DebtRecoveryStartTime* of 0xFFFFFFFF shall indicate that the value of this parameter shall remain unchanged on the Metering device following receipt of this command.

DebtRecoveryCollectionTime (mandatory): The format and use of this field is the same as for the *DebtRecoveryCollectionTime#N* attribute as defined in D.7.2.2.3.5. A *DebtRecoveryCollectionTime* of 0xFFFF shall indicate that the value of this parameter shall remain unchanged on the Metering device following receipt of this command.

DebtRecoveryFrequency (mandatory): The format and use of this field is the same as for the *DebtRecoveryFrequency#N* attribute as defined in D.7.2.2.3.6. A *DebtRecoveryFrequency* of 0xFF shall indicate that the value of this parameter shall remain unchanged on the Metering device following receipt of this command. Note that the value of this field is unused when the *DebtRecoveryMethod* is set to *Percentage Based*.

DebtRecoveryAmount (mandatory): The format and use of this field is the same as for the *DebtRecoveryAmount#N* attribute as defined in D.7.2.2.3.7. A *DebtRecoveryAmount* of 0xFFFFFFFF shall indicate that the value of this parameter shall remain unchanged on the Metering device following receipt of this command.

DebtRecoveryBalancePercentage (mandatory): The format and use of this field is the same as for the *DebtRecoveryTopUpPercentage#N* attribute as defined in D.7.2.2.3.8. A

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

DebtRecoveryBalancePercentage of 0xFFFF shall indicate that the value of this parameter shall remain unchanged on the Metering device following receipt of this command.

D.7.2.3.3.3 When Generated

This command is generated when there is a change to the debt, which the Head End System requires to be sent down to the meter.

D.7.2.3.4 **Emergency Credit Setup Command**

This command provides a method to set up the parameters for the Emergency Credit.

D.7.2.3.4.1 Payload Format

Octets	**4**	**4**	**4**	**4**
Data Type	Unsigned 32-bit Integer	UTC Time	Unsigned 32-bit Integer	Unsigned 32-bit Integer
Field Name	Issuer Event ID (M)	Start Time (M)	Emergency Credit Limit (M)	Emergency Credit Threshold (M)

Figure D-117– Emergency Credit Setup Command Payload

D.7.2.3.4.2 Payload Details

Issuer Event ID (mandatory): Unique identifier generated by the commodity provider. When new information is provided that replaces older information for the same time period, this field allows devices to determine which information is newer. The value contained in this field is a unique number managed by upstream servers or a UTC based time stamp (UTCTime data type) identifying when the command was issued. Thus, newer information will have a value in the *Issuer Event ID* field that is larger than older information.

Start Time (mandatory): A UTC Time field to denote the time at which the Emergency Credit settings become valid. A start date/time of 0x00000000 shall indicate that the command should be executed immediately. A start date/time of 0xFFFFFFFF shall cause an existing but pending *Emergency Credit Setup* command with the same *Issuer Event ID* to be cancelled.

Emergency Credit Limit (allowance) (mandatory): An unsigned 32-bit field to denote the Emergency Credit limit on the Metering Device, measured in base unit of *Currency* (as per the Price cluster) or in Units (as per the Metering cluster) with the decimal point located as indicated by the *TrailingDigits* field, as defined in the Price cluster. When no Emergency Credit has been used, this is the value defined within the *EmergencyCreditRemaining* attribute (D.7.2.2.1.3).

Emergency Credit Threshold (mandatory): An unsigned 32-bit field to denote the amount of credit remaining on the Metering Device below which the Emergency Credit facility can be selected. The value is measured in base unit of *Currency* (as per the Price cluster) or in Units (as

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

per the Metering cluster) with the decimal point located as indicated by the *TrailingDigits* field, as defined in the Price cluster.

D.7.2.3.4.3 When Generated

The *Emergency Credit Setup* command is used when the Head End System has a requirement to change the Prepayment configuration on the meter.

D.7.2.3.5 **Consumer Top Up Command**

The *Consumer Top Up* command is used by the IHD and the ESI as a method to apply credit top up values to a prepayment meter.

D.7.2.3.5.1 Payload Format

Octets	**1**	**1-26**
Data Type	8 bit Enumeration	Octet String
Field Name	Originating Device (M)	TopUp Code (M)

Figure D-118– Consumer Top Up Command Payload

D.7.2.3.5.2 Payload Details

Originating Device (mandatory): An 8 bit enumeration field identifying the Smart Energy device issuing the selection command, as defined in Table D-146.

Top Up Code (mandatory): An octet string of between 1 and 26 characters (the first character indicates the string length).

D.7.2.3.5.3 When Generated

The *Consumer Top Up* command shall be generated when a new Top-up amount of credit has been purchased from the energy supplier and is required to be sent to the Meter. Alternatively, the command can be used to transfer an instruction such as to connect or disconnect the supply, enable a particular display sequence, or other action via an appropriate *Top Up* (UTRN) *Code*.

D.7.2.3.5.4 Effect on Receipt

The meter shall update the *Top Up Date/Time#1, Top Up Amount#1* and the *Originating Device#1* attributes on the valid processing of this command. It shall then send the *ConsumerTopUpResponse* command to all devices bound to the cluster.

A Mirroring device receiving this command shall return a ZCL Default Response with a status code of NOTIFICATION_PENDING. If the command buffer on the mirror is already full, the mirror shall instead return a ZCL Default Response to the initiating device with a status code of INSUFFICIENT_SPACE. The Mirroring device may timeout the buffered message, in which

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

case it shall return a ZCL Default Response with a status code of TIMEOUT (see D.3.4.4.3 for further details).

D.7.2.3.6 Credit Adjustment Command

The *Credit Adjustment* command is sent to update the *Credit Remaining* attribute on a Prepayment meter. It shall only be sent from an ESI to the Meter.

D.7.2.3.6.1 Payload Format

Octets	**4**	**4**	**1**	**4**
Data Type	Unsigned 32-bit Integer	UTC Time	8 bits Enumeration	Signed 32-bit Integer
Field Name	Issuer Event ID (M)	Start Time (M)	Credit Adjustment Type (M)	Credit Adjustment Value (M)

Figure D-119– Credit Adjustment Command Payload

D.7.2.3.6.2 Payload Details

Issuer Event ID (mandatory): Unique identifier generated by the commodity provider. When new information is provided that replaces older information for the same time period, this field allows devices to determine which information is newer. The value contained in this field is a unique number managed by upstream servers or a UTC based time stamp (UTCTime data type) identifying when the command was issued. Thus, newer information will have a value in the *Issuer Event ID* field that is larger than older information.

Start Time (mandatory): A UTC Time field to denote the time at which the credit adjustment settings become valid. A start date/time of 0x00000000 shall indicate that the command should be executed immediately. A start date/time of 0xFFFFFFFF shall cause an existing but pending *Credit Adjustment* command with the same *Issuer Event ID* to be cancelled.

Credit Adjustment Type (mandatory): An 8-bit enumeration field identifying the type of credit adjustment to be issued out within this command. The Types are detailed within Table D-148 below.

Table D-148– Credit Type Field Enumerations

Enumerated Value	**Credit Type**
0x00	Credit Incremental
0x01	Credit Absolute
0x02 – 0xFF	Reserved

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Credit Adjustment Value (mandatory): A signed 32-bit field to denote the value of the credit adjustment, measured in base unit of *Currency* (as per the Price cluster) or in Units (as per the Metering cluster) with the decimal point located as indicated by the *TrailingDigits* field, as defined in the Price cluster. This can be a positive or negative value.

D.7.2.3.6.3 When Generated

The *Credit Adjustment* command shall be sent to the meter when the ESI has a new credit adjustment value for the meter.

D.7.2.3.6.4 Effect on Receipt

The *Credit Adjustment Value* shall be used to update the *Credit Remaining* attribute to the correct value.

D.7.2.3.7 Change Payment Mode Command

This command is sent to a Metering Device to instruct it to change its mode of operation, e.g. from Credit to Prepayment.

D.7.2.3.7.1 Payload Format

Octets	**4**	**4**	**4**	**2**	**4**
Data Type	Unsigned 32 bit Integer	Unsigned 32 bit Integer	UTC Time	16 bit BitMap	Signed 32-bit Integer
Field Name	Provider ID (M)	Issuer Event ID (M)	Implementation Date/Time (M)	Proposed Payment Control Configuration (M)	Cut Off Value (M)

Figure D-120– Change Payment Mode Command Payload

D.7.2.3.7.2 Payload Details

Provider ID (mandatory): An unsigned 32 bit field containing a unique identifier for the commodity supplier to whom this command relates.

Issuer Event ID (mandatory): Unique identifier generated by the commodity provider. When new information is provided that replaces older information for the same time period, this field allows devices to determine which information is newer. The value contained in this field is a unique number managed by upstream servers or a UTC based time stamp (UTCTime data type) identifying when the command was issued. Thus, newer information will have a value in the *Issuer Event ID* field that is larger than older information.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Implementation Date/Time (mandatory): A UTC Time field to indicate the date from which the payment mode change is to be applied. An *Implementation Date/Time* value of 0x00000000 shall indicate that the command should be executed immediately. An *Implementation Date/Time* value of 0xFFFFFFFF shall cause an existing but pending *Change Payment Mode* command with the same *Provider ID* and *Issuer Event ID* to be cancelled.

Proposed Payment Control Configuration (mandatory): An 16-bit BitMap indicating the actions required in relation to switching the payment mode. Bit encoding of this field is outlined in Table D-130.

Cut off Value (mandatory): The format and use of this field is the same as for the *CutOffValue* attribute as defined in D.7.2.2.1.19. A *CutOffValue* of 0xFFFFFFFF shall indicate that the value of this parameter shall remain unchanged on the Metering device following receipt of this command.

D.7.2.3.7.3 When Generated

The *Change Payment Mode* command shall be sent from the Energy Supplier, via the ESI, only when the need to change the mode of the meter arises.

D.7.2.3.7.4 Effect on Receipt

On receipt of the *ChangePaymentMode* command, the meter shall send the *ChangePaymentModeResponse.* The meter should create all snapshots required before the mode is changed and transmit these to the ESI. It should then also create all required snapshots and request valid Price, TOU and Prepayment information (refer to sections D.3.2.3.1.7 and D.7.2.4.2 for further details).

D.7.2.3.8 Get Prepay Snapshot Command

This command is used to request the cluster server for snapshot data.

D.7.2.3.8.1 Payload Format

Octets	**4**	**4**	**1**	**4**
Data Type	UTC Time	UTC Time	Unsigned 8-bit Integer	32-bit BitMap
Field Name	Earliest Start Time (M)	Latest End Time (M)	Snapshot Offset (M)	Snapshot Cause (M)

Figure D-121– Get Prepay Snapshot Command Payload

D.7.2.3.8.2 Payload Details

Earliest Start Time (mandatory): A UTC Timestamp indicating the earliest time of a snapshot to be returned by a corresponding *Publish Prepay Snapshot* command. Snapshots with a time stamp equal to or greater than the specified *Earliest Start Time* shall be returned.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Latest End Time (mandatory): A UTC Timestamp indicating the latest time of a snapshot to be returned by a corresponding *Publish Prepay Snapshot* command. Snapshots with a time stamp less than the specified *Latest End Time* shall be returned.

Snapshot Offset (mandatory): Where multiple snapshots satisfy the selection criteria specified by the other fields in this command, this field identifies the individual snapshot to be returned. An offset of zero (0x00) indicates that the first snapshot satisfying the selection criteria should be returned, 0x01 the second, and so on.

Snapshot Cause (mandatory): This field is used to request only snapshots for a specific cause. The allowable values are listed in Table D-151. Setting the type to 0xFFFFFFFF indicates that all snapshots should be transmitted, irrespective of the cause.

D.7.2.3.8.3 Effect on Receipt

On receipt of this command, the server will respond with the appropriate data as detailed in sub-clause D.7.2.4.2.

A ZCL Default Response with status NOT_FOUND shall be returned if the server does not have a snapshot which satisfies the received parameters (e.g. no snapshot with a timestamp between the *Earliest Start Time* and the *Latest End Time*).

D.7.2.3.9 Get Top Up Log

This command is sent to the Metering Device to retrieve the log of Top Up codes received by the meter.

D.7.2.3.9.1 Payload Format

Octets	**4**	**1**
Data Type	UTC Time	Unsigned 8-bit Integer
Field Name	Latest EndTime (M)	Number of Records(M)

Figure D-122– Get Top Up Code Log Command Payload

D.7.2.3.9.2 Payload Details

Latest End Time (mandatory): UTC timestamp indicating the latest *TopUp Time* of Top Up records to be returned by the corresponding *Publish Top Up Log* commands. The first returned Top Up record shall be the most recent record with its *TopUp Time* equal to or older than the *Latest End Time* provided.

Number of Records (mandatory): An 8-bit integer which represents the maximum number of records that the client is willing to receive in response to this command. A value of 0 would

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

11085 indicate all available records shall be returned. The first returned Top Up record shall be the
11086 most recent one in the log.

11087 D.7.2.3.9.3 Effect on Receipt

11088 On receipt of this command, the server will respond with *Publish Top Up Log* commands
11089 satisfying the specified criteria, as detailed in sub-clause D.7.2.4.5.

11090 A ZCL Default Response with status NOT_FOUND shall be returned if the server does not have
11091 any Top Up records which satisfy the received parameters (e.g. *TopUp Time* later than the
11092 *Latest End Time* provided).

11093

11094 D.7.2.3.10 **Set Low Credit Warning Level**

11095 This command is sent from client to a Prepayment server to set the warning level for low credit.

11096 D.7.2.3.10.1 Payload Format

Octets	**4**
Data Type	Unsigned 32-bit Integer
Field Name	Low Credit Warning Level (M)

11097 **Figure D-123– Set Low Credit Warning Level Command Payload**

11098 D.7.2.3.10.2 Payload Details

11099 **Low Credit Warning Level (mandatory):** An unsigned 32 bit integer that defines the consumer
11100 Low Credit value, in base unit of *Currency* (as per the Price cluster) or in Units (as per the
11101 Metering cluster), below which Low Credit warning should sound. The Low Credit warning shall
11102 be triggered when the credit remaining on the meter falls below thevalue of the *Low Credit*
11103 *Warning Level* above the disconnection point; this shall trigger the Low Credit Warning alert
11104 within this cluster.

11105

11106 D.7.2.3.11 **Get Debt Repayment Log Command**

11107 This command is used to request the contents of the Repayment log.

11108 D.7.2.3.11.1 Payload Format

Octets	**4**	**1**	**1**
Data Type	UTC Time	Unsigned 8-bit Integer	Unsigned 8 bit Integer
Field Name	Latest EndTime (M)	Number of Debts (M)	Debt Type

11109 **Figure D-124– GetDebtRepaymentLog Command Payload**

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

11110 D.7.2.3.11.2 Payload Details

11111 **Latest End Time (mandatory)**: UTC timestamp indicating the latest *Collection Time* of
11112 debt repayment records to be returned by the corresponding *Publish Debt Log* commands.
11113 The first returned debt repayment record shall be the most recent record with its
11114 *Collection Time* equal to or older than the *Latest End Time* provided.

11115 **Number of Debts (mandatory):** An 8-bit integer which represents the maximum number of debt
11116 repayment records that the client is willing to receive in response to this command. A value of 0
11117 would indicate all available records shall be returned. The first returned debt repayment record
11118 shall be the most recent one in the log.

11119 **Debt Type (mandatory):** An 8-bit enumeration field identifying the type of debt record(s) to be
11120 returned:
11121

Table D-149– Debt Type Field Enumerations

Enumerated Value	Debt Type
0x00	Debt 1
0x01	Debt 2
0x02	Debt 3
0x03 – 0xFE	Reserved
0xFF	All Debts

11122

11123 D.7.2.3.11.3 Effect on Receipt

11124 On receipt of this command, the server will respond with *Publish Debt Log* commands
11125 satisfying the specified criteria, as detailed in sub-clause D.7.2.4.6.

11126 A ZCL Default Response with status NOT_FOUND shall be returned if the server does not have
11127 any debt records which satisfy the received parameters (e.g. *Collection Time* later than the
11128 *Latest End Time* provided).

11129

11130 D.7.2.3.12 **Set Maximum Credit Limit**

11131 This command is sent from a client to the Prepayment server to set the maximum credit level
11132 allowed in the meter.

11133 D.7.2.3.12.1 Payload Format

Octets	**4**	**4**	**4**	**4**	**4**
Data Type	Unsigned 32 bit Integer	Unsigned 32 bit Integer	UTC Time	Unsigned 32-bit Integer	Unsigned 32-bit Integer

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Field Name	Provider ID (M)	Issuer Event ID (M)	Implementation Date/Time (M)	Maximum Credit Level (M)	Maximum Credit Per Top Up (M)

11134 **Figure D-125– Set Maximum Credit Level Command Payload**

11135 D.7.2.3.12.2 Payload Details

11136 **Provider ID (mandatory):** An unsigned 32 bit field containing a unique identifier for the
11137 commodity supplier to whom this command relates.

11138 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. When
11139 new information is provided that replaces older information for the same time period, this field
11140 allows devices to determine which information is newer. The value contained in this field is a
11141 unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)
11142 identifying when the command was issued. Thus, newer information will have a value in the
11143 *Issuer Event ID* field that is larger than older information.

11144 **Implementation Date/Time (mandatory):** A UTC Time field to indicate the date from which
11145 the maximum credit level is to be applied. An *Implementation Date/Time* of 0x00000000 shall
11146 indicate that the command should be executed immediately. An *Implementation Date/Time* of
11147 0xFFFFFFFF shall cause an existing but pending *Set Maximum Credit Limit* command with the
11148 same *Provider ID* and *Issuer Event ID* to be cancelled.

11149 **Maximum Credit Level (mandatory):** An unsigned 32 bit integer value indicating the
11150 maximum credit balance allowed on a meter. Any further top-up amount that will cause the
11151 meter to exceed this limit will be rejected. This value can be stated in currency (as per the Price
11152 cluster) or in units (unit of measure will be defined in the Metering cluster) depending on the
11153 Prepayment mode of operation defined in section D.7.2.2.1.1 (*Payment Control Configuration*
11154 attribute). A value of 0xFFFFFFFF will indicate that this limit is to be disabled and that all
11155 further top-ups should be permitted.

11156 **MaximumCreditPerTopUp (mandatory):** An unsigned 32-bit integer value indicating the
11157 maximum credit per top-up. Any single top-up greater than this threshold will cause the meter to
11158 reject the top-up. This value can be stated in currency (as per the Price cluster) or in units (unit of
11159 measure will be defined in the Metering cluster) depending on the Prepayment mode of operation
11160 defined in section D.7.2.2.1.1 (*Payment Control Configuration* attribute). A value of
11161 0xFFFFFFFF will indicate that this parameter is to be disabled and that there should be no limit
11162 on the amount of credit allowed in a top-up.

11163

11164 D.7.2.3.13 **Set Overall Debt Cap**

11165 This command is sent from a client to the Prepayment server to set the overall debt cap allowed
11166 in the meter.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

11167 D.7.2.3.13.1 Payload Format

Octets	4	4	4	4
Data Type	Unsigned 32 bit Integer	Unsigned 32 bit Integer	UTC Time	Signed 32-bit Integer
Field Name	Provider ID (M)	Issuer Event ID (M)	Implementation Date/Time (M)	Overall Debt Cap

11168 **Figure D-126– Set Overall Debt cap Command Payload**

11169 D.7.2.3.13.2 Payload Details

11170 **Provider ID (mandatory):** An unsigned 32 bit field containing a unique identifier for the
11171 commodity supplier to whom this command relates.

11172 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. When
11173 new information is provided that replaces older information for the same time period, this field
11174 allows devices to determine which information is newer. The value contained in this field is a
11175 unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)
11176 identifying when the command was issued. Thus, newer information will have a value in the
11177 *Issuer Event ID* field that is larger than older information.

11178 **Implementation Date/Time (mandatory):** A UTC Time field to indicate the date from which
11179 the overall debt cap is to be applied. An *Implementation Date/Time* of 0x00000000 shall indicate
11180 that the command should be executed immediately. An *Implementation Date/Time* of
11181 0xFFFFFFFF shall cause an existing but pending *Set Overall Debt Cap* command with the same
11182 *Provider ID* and *Issuer Event ID* to be cancelled.

11183 **Overall Debt Cap :** A signed 32 bit integer that defines the total amount of debt that can be
11184 taken from top-ups (in the case of multiple instantiated top-up based debts on the Metering
11185 Device) (see D.7.2.2.1.7). This field is always a monetary value, and as such the field is
11186 measured in a base unit of *Currency* with the decimal point located as indicated by the *Trailing*
11187 *Digits* field, as defined in the Price cluster.

11188

11189 ### D.7.2.4 Commands Generated

11190 Table D-150 lists commands that are generated by the server.

11191 **Table D-150– Cluster -specific Commands Sent by the Server**

Command Identifier Field Value	Description	Mandatory/ Optional
0x00	*Reserved*	O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x01	*Publish Prepay Snapshot*	O
0x02	*Change Payment Mode Response*	O
0x03	*Consumer Top Up Response*	O
0x04	*Reserved*	O
0x05	*Publish Top Up Log*	O
0x06	*Publish Debt Log*	O

11192

11193 D.7.2.4.1 **Supply Status Response Command**

11194 The *Supply Status Response* command has been moved to the Metering cluster (see Annex D.3
11195 for further details).

11196

11197 D.7.2.4.2 **Publish Prepay Snapshot Command**

11198 This command is generated in response to a *GetPrepaySnapshot* command or when a new
11199 snapshot is created. It is used to return a single snapshot to the client.

11200 D.7.2.4.2.1 Payload Format

Octets	**4**	**4**	**1**	**1**	**1**	**4**	**1**	**Variable**
Data Type	Unsigned 32-bit Integer	UTC Time	Unsigned 8-bit Integer	Unsigned 8-bit Integer	Unsigned 8-bit Integer	32-bit BitMap	8-bit Enumeration	
Field Name	Snapshot ID (M)	Snapshot Time (M)	Total Snapshots Found (M)	Command Index (M)	Total Number of Commands (M)	Snapshot Cause (M)	Snapshot Payload Type (M)	Snapshot Payload (M)

11201 **Figure D-127– Publish Prepay Snapshot Command Payload**

11202 D.7.2.4.2.2 Payload Details

11203 **Snapshot ID (mandatory):** Unique identifier allocated by the device creating the snapshot.

11204 **Snapshot Time (mandatory):** This is a 32 bit value (in UTC Time) representing the time at
11205 which the data snapshot was taken.

11206 **Total Snapshots Found (mandatory):** An 8-bit Integer indicating the number of snapshots
11207 found, based on the search criteria defined in the associated *GetPrepaySnapshot* command. If the
11208 value is greater than 1, the client is able to request the next snapshot by incrementing the
11209 *Snapshot Offset* field in an otherwise repeated *GetPrepaySnapshot* command.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

11210 **Command Index (mandatory):** The *Command Index* is uses to count the payload fragments in
11211 the case where the entire payload does not fit into one message. The *Command Index* starts at 0
11212 and is incremented for each fragment belonging to the same command.

11213 **Total Number of Commands (mandatory):** In the case where the entire payload does not fit
11214 into one message, the *Total Number of Commands* field indicates the total number of sub-
11215 commands in the message.

11216 **Snapshot Cause (mandatory):** A 32-bit BitMap indicating the cause of the snapshot. The
11217 snapshot cause values are listed in Table D-151.

11218 **Table D-151– Snapshot Payload Cause**

Bit	Description
0	General
1	End of Billing Period
2	Reserved for Metering cluster
3	Change of Tariff Information
4	Change of Price Matrix
5	Reserved for Metering cluster
6	Reserved for Metering cluster
7	Reserved for Metering cluster
8	Reserved for Metering cluster
9	Reserved for Metering cluster
10	Manually Triggered from Client
11	Reserved for Metering cluster
12	Change of Tenancy
13	Change of Supplier
14	Change of Meter Mode
15	Reserved for Metering cluster
16	Reserved for Metering cluster
17	Reserved for Metering cluster
18	TopUp addition
19	Debt/Credit addition
20-31	Reserved

11219 NOTE: Where applicable, these Prepayment snapshots shall be taken in conjunction with the
11220 associated snapshots in the Metering cluster.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

SnapshotPayloadType (mandatory): The *SnapshotPayloadType* is an 8-bit enumerator defining the format of the *SnapshotPayload* in this message. The different snapshot types are listed in Table D-152. The server selects the *SnapshotPayloadType* based on the charging scheme in use.

Table D-152– Snapshot Payload Type

Enumeration	Description
0x00	Debt/Credit Status
0x01 – 0xFE	Reserved
0xFF	Not used

SnapshotPayload (mandatory): the format of the *SnapshotPayload* differs depending on the *SnapshotPayloadType*.

D.7.2.4.2.2.1 SnapshotPayloadType = Debt/Credit Status

Octets	4	4	4	4	4	4
Data Type	Signed 32-bit Integer	Unsigned 32-bit Integer	Unsigned 32-bit Integer	Unsigned 32-bit Integer	Signed 32-bit Integer	Signed 32-bit Integer
Field Name	Accumulated Debt (M)	Type 1 Debt Remaining (M)	Type 2 Debt Remaining (M)	Type 3 Debt Remaining (M)	Emergency Credit Remaining (M)	Credit Remaining (M)

Figure D-128– Debt/Credit Status Command Payload

Accumulated Debt (mandatory): The *AccumulatedDebt* field represents the total amount of debt remaining on the Metering Device, measured in a base unit of *Currency* with the decimal point located as indicated by the *Trailing Digits* field, as defined in the Price cluster.

Type 1 Debt Remaining (mandatory): The *Type1DebtRemaining* field represents the amount of Type 1 debt remaining on the Metering Device, measured in base unit of *Currency* with the decimal point located as indicated by the *TrailingDigits* field, as defined in the Price cluster.

Type 2 Debt Remaining (mandatory): The *Type2DebtRemaining* field represents the amount of Type 2 debt remaining on the Metering Device, measured in base unit of *Currency* with the decimal point located as indicated by the *TrailingDigits* field, as defined in the Price cluster.

Type 3 Debt Remaining (mandatory): The *Type3DebtRemaining* field represents the amount of Type 3 debt remaining on the Metering Device, measured in base unit of *Currency* with the decimal point located as indicated by the *TrailingDigits* field, as defined in the Price cluster.

Emergency Credit Remaining (mandatory): The *EmergencyCreditRemaining* field represents the amount of Emergency Credit still available on the Metering Device. If Monetary based, this field is measured in a base unit of *Currency* (as per the Price cluster) or in Units (as per the

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Metering cluster), with the decimal point located as indicated by the *TrailingDigits* field, as defined in the Price cluster. If Unit based, the unit of measure is as defined in the Metering cluster (see sub-clause D.3.2.2.4.1).

Credit Remaining (mandatory): The *CreditRemaining* field represents the amount of credit remaining on the Metering Device. If Monetary based, this field is measured in a base unit of Currency (as per the Price cluster) or in Units (as per the Metering cluster), with the decimal point located as indicated by the *TrailingDigits* field, as defined in the Price cluster. If Unit based, the unit of measure is as defined in the Metering cluster (see sub-clause D.3.2.2.4.1).

D.7.2.4.3 Change Payment Mode Response Command

This command is sent in response to the *ChangePaymentMode* command. The *ChangePaymentModeResponse* command shall only inform the ESI of the current default setting that would affect the meter when entering into Prepayment/PAYG or Credit mode. Should these values require changing then other commands within the Prepayment & Price cluster should be used.

D.7.2.4.3.1 Payload Format

Octets	**1**	**4**	**4**	**4**
Data Type	8 Bit Bitmap	Unsigned 32-bit Integer	Unsigned 32-bit Integer	Unsigned 32-bit Integer
Field Name	Friendly Credit (M)	Friendly Credit Calendar ID (M)	Emergency Credit Limit (M)	Emergency Credit Threshold (M)

Figure D-129– Change Payment Mode Response Command Payload

D.7.2.4.3.2 Payload Details

Friendly Credit (mandatory): An 8-bit BitMap to show if the meter has a Friendly Credit calendar and that this calendar shall be enabled.

Table D-153– Friendly Credit BitMap

Bit	**Description**
0	Friendly credit enabled
1-7	Reserved

Friendly Credit Calendar ID (mandatory): An unsigned 32-bit field to denote the *IssuerCalendarID* that shall be used for the friendly credit periods. The *IssuerCalendarID* can be found within the TOU cluster (see Annex D.9).

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Emergency Credit Limit/Allowance (mandatory): An unsigned 32-bit field to denote the emergency credit limit on the Metering Device, measured in base unit of *Currency* with the decimal point located as indicated by the *TrailingDigits* field, as defined in the Price cluster. Should no emergency credit have been used, this is the value defined within the *EmergencyCreditRemaining* attribute (D.7.2.2.1.3).

Emergency Credit Threshold (mandatory): An unsigned 32-bit field to denote the amount of credit remaining on the Metering Device below which the *Emergency Credit* facility can be selected. The value is measured in base unit of *Currency* with the decimal point located as indicated by the *TrailingDigits* field, as defined in the Price cluster.

D.7.2.4.3.3 When Generated

The *ChangePaymentModeResponse* command is generated in response to a *ChangePaymentMode* command.

D.7.2.4.4 Consumer Top Up Response Command

The Metering device responds either with the following values in the case of a credit token received:

- Meter's enumerated status, after receiving the top up, in the *Result Type* field
- Received Top up token's credit value in the *Top Up Value* field
- The source of the top up, enumerated in the *Source of Top up* field
- The credit remaining on the meter, after the addition of this Top Up, in the *Credit Remaining* field,

OR, in the case of a connect/disconnect Top Up (UTRN) code, with the following:

- Supply status, after processing of the token, enumerated in the *Result Type* field
- Top up token's credit value SET TO ZERO in the *Top Up Value* field
- The source of the top up, enumerated in the *Source of Top up* field
- The credit remaining on the meter, after the addition of this Top Up, in the *Credit Remaining* field

D.7.2.4.4.1 Payload Format

Octets	**1**	**4**	**1**	**4**
Data Type	8-bit Enumeration	Signed 32- bit Integer	8-bit Enumeration	Signed 32-bit Integer
Field Name	Result Type (M)	Top Up Value (M)	Source of Top up (M)	Credit Remaining (M)

Figure D-130– Consumer Top Up Response Command Payload

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

11300 D.7.2.4.4.2 Payload Details

11301 **Result Type (mandatory):** An 8-bit enumerated value indicating whether the Metering Device
11302 accepted or rejected the top up. Enumerated values are described in Table D-154

11303 **Table D-154– Result Type Field Enumerations**

Enumerated Value	Result Type Description
0x00	Accepted
0x01	Rejected-Invalid Top Up
0x02	Rejected-Duplicate Top Up
0x03	Rejected-Error
0x04	Rejected-Max Credit Reached
0x05	Rejected-Keypad Lock
0x06	Rejected-Top Up Value Too Large
0x07 – 0x0F	Reserved
0x10	Accepted – Supply Enabled
0x11	Accepted – Supply Disabled
0x12	Accepted – Supply Armed
0x13 – 0xFF	Reserved

11304

11305 **Top up Value (mandatory):** A signed 32-bit integer field representing the Top Up value
11306 available in the top up content. If it is Monetary based, this field is measured in a base unit of
11307 *Currency* with the decimal point located as indicated by the Trailing Digits field, as defined in
11308 the Price cluster. If Unit based, the unit of measure is as defined in the Metering cluster (see sub-
11309 clause D.3.2.2.4.1). If *Result Type* is other than *Accepted*, this field has a maximum value
11310 (0xFFFFFFFF) which indicates an invalid Top Up value.

11311 **Source of Top Up (mandatory):** An 8-bit enumeration indicating the device that has issued the
11312 top up (see Table D-146 for applicable enumerations).

11313 **Credit Remaining (mandatory):** The *Credit Remaining* field represents the amount of credit
11314 remaining on the Metering Device after addition of a top up. If Monetary based, this field is
11315 measured in a base unit of *Currency* with the decimal point located as indicated by the *Trailing*
11316 *Digits* field, as defined in the Price cluster. If Unit based, the unit of measure is as defined in the
11317 Metering cluster (see sub-clause D.3.2.2.4.1). In case of *Result Type* other than *Accepted*, the
11318 *Credit Remaining* field has a maximum value (0xFFFFFFFF) representing invalid credit
11319 remaining.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

11320 D.7.2.4.4.3 When Generated

11321 The *ConsumerTopUpResponse* command is generated in response to a *ConsumerTopUp*
11322 command.

11323

11324

11325 D.7.2.4.5 **Publish Top Up Log Command**

11326 This command is used to send the Top Up Code Log entries to the Prepayment client. The
11327 command shall be sent in response to a *Get Top Up Log* command and MAY be sent
11328 unsolicited whenever a new Top Up code is received and successfully processed[19]. When the
11329 command is being sent a the result of a Top Up, the *Top Up Payload* shall contain details for that
11330 Top Up only. Where the *Top Up Payload* contains details for more than one log entry, they are
11331 sent most recent entry first.

11332 D.7.2.4.5.1 Payload Format

Octets	**1**	**1**	**xx**
Data Type	Unsigned 8 –bit Integer	Unsigned 8 –bit Integer	
Field Name	Command Index (M)	Total Number of Commands (M)	Top Up Payload

11333 **Figure D-131– Publish Top Up Log Command Payload**

11334 D.7.2.4.5.2 Payload Details

11335 **Command Index (mandatory):** The *Command Index* is used to count the payload fragments in
11336 the case where the entire payload does not fit into one message. The *Command Index* starts at 0
11337 and is incremented for each fragment belonging to the same command. A value of 0xFE in the
11338 *Command Index* states that this is the last expected command.

11339 **Total Number of Commands (mandatory):** In the case that an entire payload does not fit into
11340 one message, the *Total Number of Commands* field indicates the total number of sub-commands
11341 in the message.

11342 ***D.7.2.4.5.2.1 Top Up Payload Details***

Octets	**1..26**	**4**	**4**	**1..26**	**4**	**4**	**1..26**	**4**	**4**
Data Type	Octet String	Signed 32-bit Integer	UTC Time	Octet String	Signed 32-bit Integer	UTC Time	Octet String	Signed 32-bit Integer	UTC Time
Field Name	TopUp Code (M)	TopUp Amount (M)	TopUp Time (M)	TopUp Code +1 (M)	TopUp Amount + 1 (M)	TopUp Time + 1 (M)	TopUp Code +n (M)	TopUp Amount + n (M)	TopUp Time + n (M)

11343 **Figure D-132– Top Up Code Payload**

[19] CCB 2009

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

11344 **TopUp Code (mandatory):** This is the value of the Top Up code stored in the log.

11345 **TopUp Amount (mandatory):** This is the amount of credit that was added to the Metering
11346 Device during this Top Up.

11347 **TopUp Time (mandatory):** This is the UTC Timestamp when the Top Up was applied to the
11348 Metering Device.

11349

11350 D.7.2.4.6 **Publish Debt Log Command**

11351 This command is used to send the contents of the Repayment Log.

11352 D.7.2.4.6.1 Payload Format

Octets	1	1	xx
Data Type	Unsigned 8 –bit Integer	Unsigned 8 –bit Integer	
Field Name	Command Index (M)	Total Number of Commands (M)	Debt Payload (M)

11353 **Figure D-133– Publish Debt Log Command Payload**

11354 D.7.2.4.6.2 Payload Details

11355 **Command Index (mandatory):** The *Command Index* is used to count the payload fragments in
11356 the case where the entire payload does not fit into one message. The *Command Index* starts at 0
11357 and is incremented for each fragment belonging to the same command. A value of 0xFE in the
11358 *Command Index* states that this is the last expected command.

11359 **Total Number of Commands (mandatory):** In the case that an entire payload does not fit into
11360 one message, the *Total Number of Commands* field indicates the total number of sub-commands
11361 in the message.

11362 **Debt Payload (mandatory):** The *Debt Payload* shall contain one or more debt records, each of
11363 which shall be of the following format:-

Octets	4	4	1	4
Data Type	UTC Time	Unsigned 32-bit Integer	8-bit Enumeration	Unsigned 32-bit Integer
Field Name	Collection Time (M)	Amount Collected (M)	Debt Type (M)	Outstanding Debt (M)

11364 **Figure D-134– Format of a Debt Payload Record**

11365 **Collection Time (mandatory):** An UTC time field identifying the time when the collection
11366 occurred.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Amount Collected (mandatory): An unsigned 32-bit field to denote the amount of debt collected at this time. This parameter shall be measured in base unit of *Currency* with the decimal point located as indicated by the *Trailing Digits* field, as defined in the Price cluster.

Debt Type (mandatory): An 8-bit enumeration field identifying the type of debt the record refers to. The enumerations are defined in Table D-149.

Outstanding Debt (mandatory): An unsigned 32-bit field to denote the amount of debt still outstanding after the debt was collected. This parameter shall be measured in base unit of *Currency* with the decimal point located as indicated by the *Trailing Digits* field, as defined in the Price cluster.

D.7.3 Client

D.7.3.1 Dependencies

- Support for ZCL Data Types
- Events carried using this cluster include a timestamp with the assumption that target devices maintain a real time clock. Devices can acquire and synchronize their internal clocks via the ZCL Time server.

D.7.3.2 Attributes

The client has no attributes.

D.7.3.3 Commands Received

The client receives the cluster-specific response commands detailed in D.7.2.4.

D.7.3.4 Commands Generated

The client generates the cluster-specific commands detailed in D.7.2.3, as required by the application.

D.7.4 Application Guidelines

D.7.4.1 Credit Status Attribute

The purpose of the *Credit Status* attribute is to describe to any device on the HAN, what the status of a meter operating in Prepayment mode may be at any point in time. There are a number of important functionalities in Prepayment meters, and a variety of implementations depending on the manufacturer and their chosen system, however this attribute is designed to pick up the

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

lowest common denominator of statuses that would be important to an end user looking to glean information about their meter in the HAN. For example, has their meter run out of credit, is Emergency Credit available or has Emergency Credit been selected?

The diagram below describes the manner in which this attribute SHOULD be used when describing these statuses and others. This guidance note is not designed to prescribe how any Prepayment meter logic works, but merely to get a common understanding of the meter status to the end users' interface device. It is entirely up to device manufacturers to decide how to best use this information and display it.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

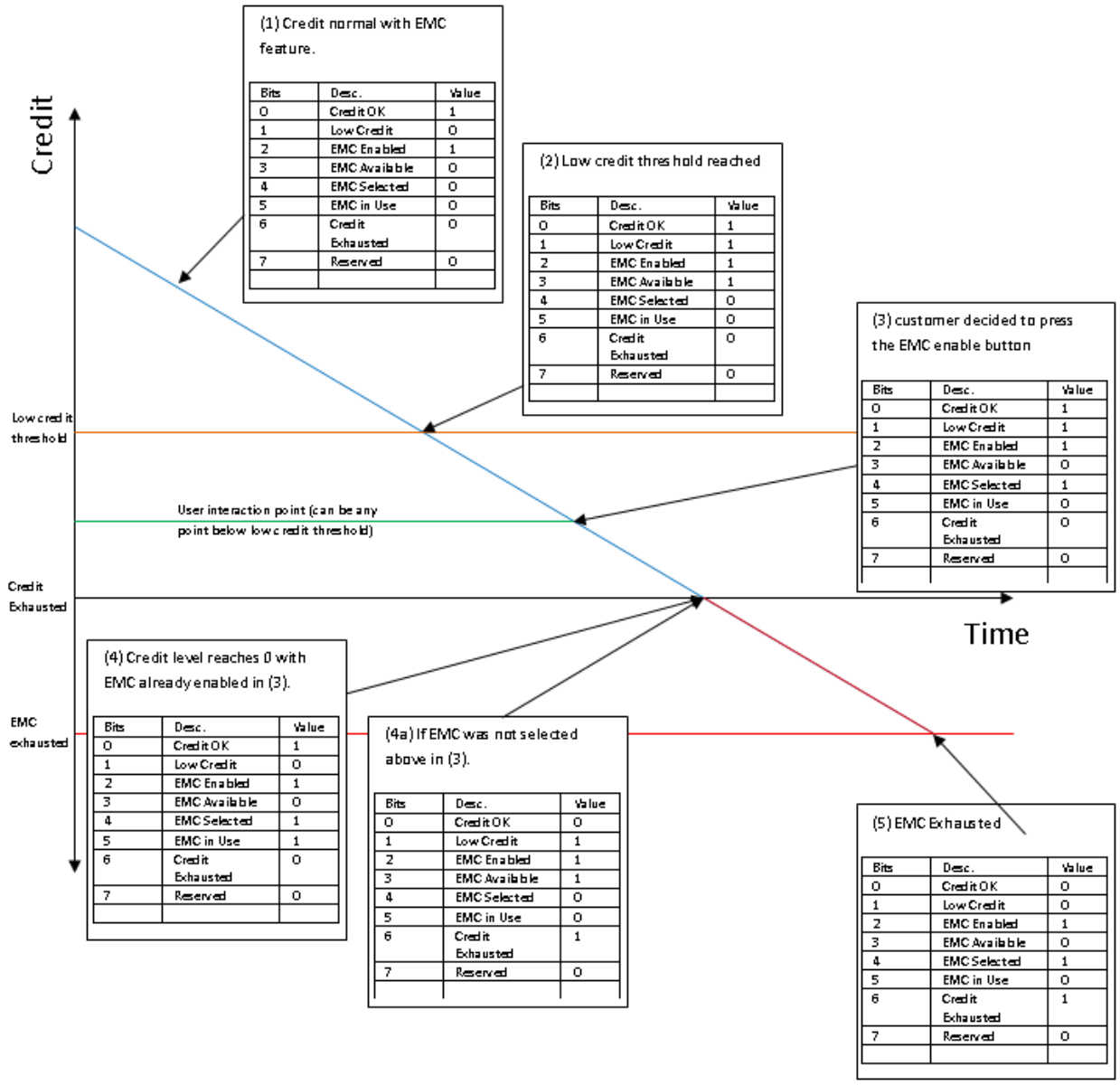


Figure D-135– Prepayment Credit Status Attribute Explained

D.7.4.1.1 Statuses Explained - an Example

Below is a brief explanation of each status noted on the diagram above in order to give a better indication of what the meter is doing at any given point. Imagine that the diagonal blue line represents the customer's credit, and when it turns into a plum colored diagonal line below the Time-Axis, the reader can assume that the meter is in negative credit, and Emergency Credit may or may not be invoked depending on the use case.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

11415 The definitions of functionality below are modeled on the current understanding of Prepayment
11416 functionality. However there could well be a situation when meters are not disconnected when
11417 reaching the zero credit point, or indeed when Emergency Credit has been exhausted. This
11418 description is designed to aid understanding only and not specify meter functionality (see Figure
11419 D-135):

11420 1. At this stage the meter has customer credit and has the Emergency Credit feature enabled.
11421 This means that when the meter reaches the Low Credit threshold, Emergency Credit will be
11422 available to be selected by the end user.

11423 2. At this point the meter still has customer credit available, but the meter has now reached the
11424 Low Credit threshold. This means that the end user may, should they choose to do so, select to
11425 engage the Emergency Credit. This will allow the meter to pass into a predefined amount of
11426 negative credit, without disconnection, when the meter credit reaches zero. The Emergency
11427 Credit can be selected at any point below the Low Credit threshold, but if this is not done
11428 before the customer's credit reaches zero then the meter will disconnect the supply.

11429 3. Same as above except this is demonstrating the point at which the end user actually engages
11430 the Emergency Credit function, and in doing so making Emergency Credit no longer available
11431 for selection again.

11432 4. Meter reaches zero credit with Emergency Credit function engaged. This means that the
11433 option to engage Emergency Credit functionality is not available to the end user (as he has
11434 already done it), but the meter is still connected and 'Credit OK' remains set because
11435 Emergency Credit is available.
11436 a. In this case the end user has decided not to engage Emergency Credit functionality
11437 before the credit level reaches zero, thereby removing the 'Credit OK' flag once the
11438 available credit has reached zero. The Emergency Credit function is still available,
11439 but requires end user interaction in order to engage it.

11440 5. At this point Emergency Credit is exhausted and the meter is assumed to have disconnected
11441 (this may not be the case depending on the supplier's requirements). There is no available
11442 credit or Emergency Credit, and it is not possible for the end user to engage the Emergency
11443 Credit function.

11444 At this point in time, when all credit is exhausted, the meter and IHD will need to display the
11445 "debt to clear". This is the amount of credit that must be put onto the meter in order to exceed
11446 the Low Credit warning threshold and get the meter back on supply, with Emergency Credit
11447 available again (credit above zero will get the lights back on but Emergency Credit will not be
11448 available until credit is above the Low Credit Warning Threshold). The 'debt to clear' will be
11449 transmitted by way of the Credit Remaining register (as it will be a negative number at this
11450 time, made up of the debt that the meter has accrued while in Emergency Credit).

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

11451 If Standing Charge, debt repayment charges and energy charges are normally being paid, these
11452 may not all be charged during an Emergency Credit period, but will still accrue in the
11453 background until Emergency Credit is exhausted (at point 5). Depending on energy supplier
11454 preference, it SHALL be configurable whether or not Emergency Credit is used to pay debt
11455 charges. The Emergency Credit value, along with debt charges accrued in the background
11456 while Emergency Credit was in operation, will be added to the 'debt to clear' register in the
11457 meter when Emergency Credit is exhausted, and displayed on the Credit Remaining register as
11458 a negative number.

11459

11460

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

11461 ## D.8 Over-the-Air Bootload Cluster

11462 ### D.8.1 Overview

11463 The over-the-air bootload cluster provides a common mechanism to manage and serve up
11464 upgrade images for devices from different manufacturers in the same network. Servers provide
11465 firmware images to clients to download, controlling the timing for downloads and when the
11466 actual upgrade to a new version of software is made. Clients periodically query the server for
11467 new images and then can download the image at a rate according to their capabilities or
11468 policies.

11469 Details for the over-the-air (OTA) bootload cluster are maintained in a separate document,
11470 reference [095264r15].

11471 Smart Energy devices may optionally support the over-the-air bootload cluster client or server.
11472 If the OTA cluster is implemented by a Smart Energy device then APS encryption on all unicast
11473 messages shall be used. Smart Energy devices that implement the client must support ECDSA
11474 signature verification of images. Either Crypto Suite 1 or Crypto Suite 2 can be used to sign the
11475 image.

11476 Additionally, over-the-air bootload cluster client devices that are intended to be field
11477 upgradeable to Smart Energy 2.0 should support the optional feature “query specific file” in
11478 order to potentially receive device specific data necessary for the transition to a Smart Energy 2.0
11479 device.

11480 ### D.8.2 OTA Bootloading Timing Considerations

11481 The OTA cluster defines the message formatting used to pass device images but does not
11482 specify when to use the cluster. The following policies specify how and when to use the OTA
11483 cluster such that all devices in an SE network will upgrade at predictable intervals.

11484 **1** OTA clients shall perform service discovery to find the OTA server after registration has
11485 completed.

11486 **2** An OTA client device that does not find an OTA server in the network shall periodically
11487 attempt a new discovery once a day.

11488 **3** All devices shall query the OTA server at least once a day for information about the next
11489 version to upgrade to. Non-sleepy devices in the network may be instructed to begin a new
11490 download at any point in time via the *Image Notify* command.

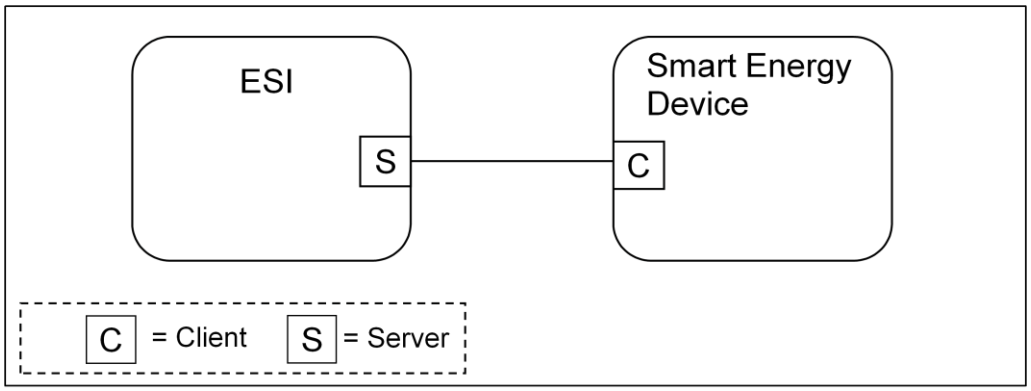
11491 **4** All client devices may download data as quickly as their capabilities allow, but at a minimum
11492 rate of one block per 10 minutes. This means that at a rate of 1 block (50 bytes) per 10
11493 minutes, a 128k file will take 18 days to download.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

11494 # D.9 Calendar Cluster

11495 ## D.9.1 Overview

11496 The Calendar cluster implements commands to transfer calendar information within the
11497 premises. The calendar information is distributed by an ESI.
11498



11499
11500 **Figure D-136–Calendar Cluster**

11501 The server shall be able to store at least **two** instances of the calendar, typically the current and
11502 the next one. It is recommended that a client is also capable of storing 2 instances. It is also
11503 recommended that a Calendar server may additionally store at least **one** previous instance of the
11504 calendar.

11505 The Calendar server shall send unsolicited *PublishCalendar* and *PublishSpecialDays* commands
11506 to its clients if they are bound to it. Other calendar items such as Day Profiles, Week Profiles and
11507 Season information shall not be sent unsolicited. The clients shall send corresponding Get…
11508 commands to fetch the information from the server as necessary. The Calendar server shall
11509 publish new calendars, to clients that have bound to receive them, as soon as they become
11510 available. Devices with limited resources, and which cannot therefore handle multiple calendars,
11511 should NOT 'register' (i.e. bind to the server) to receive unsolicited Calendar cluster commands.
11512 If there is no next calendar available, a ZCL Default Response shall be returned with status
11513 NOT_FOUND; the ESI shall publish the information as soon as it gets it from the HES. Devices
11514 (particularly battery-powered devices) should regularly check for updates to calendar items.

11515 The Calendar must be replaced as a whole; only the Special Day Table can be changed
11516 independently. To uniquely identify the parts of a calendar, an Issuer Calendar ID is used. All
11517 parts belonging to the same calendar must have the same Issuer Calendar ID. All parts of a
11518 particular calendar shall be successfully retrieved from the server before a client can use that
11519 calendar. It is anticipated that a change to any part of a calendar, other than a Special Day Table,
11520 will result in a new calendar and a new Issuer Calendar ID.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

The Calendar cluster will support all of the following calendar types:

- Delivered
- Received
- Delivered and Received
- Friendly Credit
- Auxiliary Load Switch

Each calendar has three associated tables, a Season table, a Week Profile table and a Day Profile table. These are described in Table D-155. In addition, there is a Special Day Table which allows special days to be defined (days where a special switching behavior overrides the normal operation). Each entry in the Special Day table contains a date together with the Day ID for a Day Profile (in the associated Calendar's Day Profile table) to be used on that date.

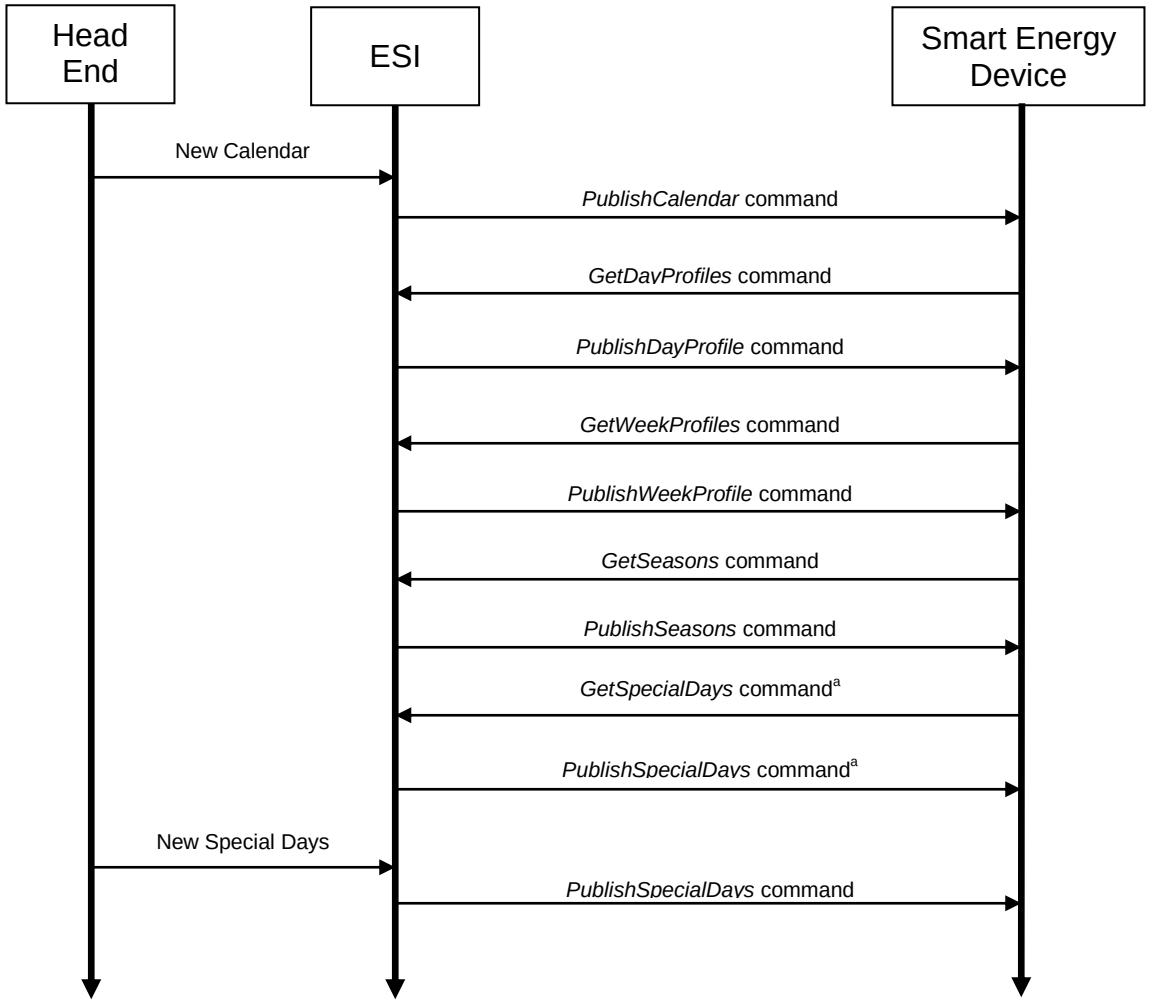
Table D-155–Calendar Data Structures

Table	Description
Season Table	Contains a list of Seasons defined by their starting date and a reference to the Week Profile to be executed. The list is arranged according to Season Start Date. The Week ID Ref defines the Week Profile active in this Season. If no season is defined, it is expected that the calendar will have one repeating Week Profile. NOTE: A 'Season', while normally considered to be a 3 or 6 month period, could be used for other arbitrary periods e.g. monthly or quarterly. The minimum resolution is 1 day, although a week would normally be the smallest interval.
Week Profile Table	Contains an array of Week Profiles to be used in the different Seasons. For each Week Profile, the Day Profile for every day of a week is identified. Monday to Sunday reference the Day ID of the Day Profile to be used for the corresponding day. The same Day Profile may be used for more than one day of the week. If no Week Profile is defined, it is expected that the calendar will have one repeating Day Profile.
Day Profile Table	Contains an array of Day Profiles, identified by their Day ID. Each Day Profile contains a list of scheduled actions and is defined by a script to be executed at the corresponding activation time (Start Time). The list is arranged according to Start Time.
Special Day Table	Defines special dates. On such dates, a special switching behavior overrides the normal one defined by the Season and Week Profile Tables. The Day Profile referenced through the Day ID in the Special Days Table activates the Day Schedule of the corresponding Day Profile.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

All dates and times shall be defined according to UTC, Standard or Local time. Alternatively, the Season Table may be used to accommodate requirements such as daylight saving.

Figure D-137 shows a recommended Calendar command sequence (noting that this sequence is for a main-powered Smart Energy Device) :



[a]Although not necessary, it is thought wise to check for updates when a new calendar is published

Figure D-137– Recommended Calendar Command Sequence

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

11573

11574 ## D.9.2 Server

11575 ### D.9.2.1 Dependencies

11576 A device implementing the Calendar server shall also implement the Price server. A device
11577 implementing the Calendar client shall also implement the Price client. The commodity type of a
11578 Calendar server shall be inferred from that of the corresponding Price server (i.e. located on the
11579 same device/endpoint). It is expected that the TOU calendar and tariff information of the Price
11580 cluster is provided by the same utility supplier. The *ProviderID* for the TOU calendar shall be
11581 obtained from the *Tariff Information Set* of the Price Cluster.

11582 ### D.9.2.2 Attributes

11583 For convenience, the attributes defined in this cluster are arranged into sets of related
11584 attributes; each set can contain up to 256 attributes. Attribute identifiers are encoded such that
11585 the most significant Octet specifies the attribute set and the least significant Octet specifies the
11586 attribute within the set. The currently defined attribute sets are listed in the following Table
11587 D-156.

11588 **Table D-156–Calendar Cluster Attribute Sets**

Attribute Set Identifier	Description
0x00	Auxiliary Switch Label Attribute Set
0x01 – 0xFF	Reserved

11589

11590 #### D.9.2.2.1 Auxiliary Switch Label Attribute Set

11591 **Table D-157– Auxiliary Switch Label Attribute Set**

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0000	AuxSwitch1Label	Octet String	1 to 23 Octets	Read/ Write	“Auxiliary 1”	O
0x0001	AuxSwitch2Label	Octet String	1 to 23 Octets	Read/ Write	“Auxiliary 2”	O
0x0002	AuxSwitch3Label	Octet String	1 to 23 Octets	Read/ Write	“Auxiliary 3”	O
0x0003	AuxSwitch4Label	Octet String	1 to 23 Octets	Read/ Write	“Auxiliary 4”	O
0x0004	AuxSwitch5Label	Octet String	1 to 23 Octets	Read/ Write	“Auxiliary 5”	O
0x0005	AuxSwitch6Label	Octet String	1 to 23 Octets	Read/ Write	“Auxiliary 6”	O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x0006	AuxSwitch7Label	Octet String	1 to 23 Octets	Read/ Write	“Auxiliary 7”	O
0x0007	AuxSwitch8Label	Octet String	1 to 23 Octets	Read/ Write	“Auxiliary 8”	O
0x0008 – 0x00FF	Reserved					

11592

11593 D.9.2.2.1.1 AuxSwitchNLabel Attributes

11594 The *AuxSwitchNLabel* attributes provide a method for assigning a label to an Auxiliary Switch.
11595 The *AuxSwitchNLabel* attributes are ZCL Octet String fields capable of storing 22-character
11596 strings (the first Octet indicates length) encoded in the UTF-8 format.
11597

11598 ### D.9.2.3 Commands Generated

11599 Table D-158 lists commands that are generated by the server.

11600 **Table D-158– Commands Generated by the Calendar Cluster Server**

Command Identifier Field Value	Description	Mandatory / Optional
0x00	PublishCalendar	M
0x01	PublishDayProfile	M
0x02	PublishWeekProfile	M
0x03	PublishSeasons	M
0x04	PublishSpecialDays	M
0x05	CancelCalendar	O

11601

11602 D.9.2.3.1 **PublishCalendar Command**

11603 The *PublishCalendar* command is published in response to a *GetCalendar* command or if new
11604 calendar information is available. The Calendar must be replaced as a whole; only the Special
11605 Day Table can be changed independently. All parts of a calendar instance shall have the same
11606 Calendar ID.

11607 Nested and overlapping calendars are not allowed. In the case of overlapping calendars of the
11608 same type (calendar type), the calendar with the newer *IssuerCalendarID* takes priority over all
11609 nested and overlapping calendars. All existing calendar instances that overlap, even partially,
11610 should be removed. The only exception to this is if a calendar instance with a newer *Issuer Event*
11611 *ID* overlaps with the end of the current active calendar but is not yet active, then the active
11612 calendar is not deleted but modified so that the active calendar ends when the new calendar
11613 begins.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

11614 D.9.2.3.1.1 Payload Format

Octets	4	4	4	4	1	1	1..13
Data Type	Unsigned 32-bit Integer	Unsigned 32-bit Integer	Unsigned 32-bit Integer	UTC Time	8-bit Enumeration	Unsigned 8-bit Integer	Octet String
Field Name	Provider Id (M)	Issuer Event ID (M)	Issuer Calendar ID (M)	Start Time (M)	Calendar Type (M)	Calendar Time Reference (M)	Calendar Name (M)

11615

Octets	1	1	1
Data Type	Unsigned 8-bit Integer	Unsigned 8-bit Integer	Unsigned 8-bit Integer
Field Name	Number of Seasons (M)	Number of Week Profiles (M)	Number of Day Profiles (M)

11616 **Figure D-138– *PublishCalendar* Command Payload**

11617 D.9.2.3.1.2 Payload Details

11618 **Provider Id (mandatory):** An unsigned 32-bit field containing a unique identifier for the
11619 commodity provider. This field allows differentiation in deregulated markets where multiple
11620 commodity providers may be available.

11621 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. When
11622 new information is provided that replaces older information for the same time period, this field
11623 allows devices to determine which information is newer. The value contained in this field is a
11624 unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)
11625 identifying when the Publish command was issued. Thus, newer information will have a value in
11626 the *Issuer Event ID* field that is larger than older information.

11627 **Issuer Calendar ID (mandatory):** Unique identifier generated by the commodity Supplier to
11628 identify a particular calendar.

11629 **Start Time (mandatory):** A UTC Time field to denote the time at which the published calendar
11630 becomes valid. A start date/time of 0x00000000 shall indicate that the command should be
11631 executed immediately.

11632 **Calendar Type (mandatory):** An 8-bit enumeration identifying the type of calendar published
11633 in this command. Table D-159 details the enumeration of this field. Generation Meters shall use
11634 the ‘Received’ Calendar.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

11635

11636 **Table D-159– Calendar Type Enumeration**

Value	Description
0x00	Delivered Calendar
0x01	Received Calendar
0x02	Delivered and Received Calendar
0x03	Friendly Credit Calendar
0x04	Auxillary Load Switch Calendar
0x05-0xFF	Reserved

11637

11638 **Calendar Time Reference (mandatory):** This field indicates how the Start Times contained in
11639 the calendar are to be interpreted. The following table shows possible values:

11640 **Table D-160– Calendar Time Reference Enumeration**

Value	Description
0x00	UTC time
0x01	Standard time
0x02	Local time
0x03-0xFF	Reserved

11641

11642 Standard time refers to UTC time adjusted according to the local time zone.

11643 Local time refers to Standard time adjusted according to local daylight savings regulations.

11644 Where the optional Standard and/or Local Time (as applicable) are not available on the Time
11645 cluster server (and are not managed locally by the meter), the *Calendar Time Reference* shall
11646 default to UTC time.

11647 **Calendar Name (mandatory):** The *CalendarName* provides a method for utilities to assign a
11648 name to the entire calendar. The *CalendarName* is a ZCL Octet String field capable of storing a
11649 12 character string (the first Octet indicates length) encoded in the UTF-8 format.

11650 **Number of Seasons (mandatory):** Number of entries in the Seasons Table. A value of 0x00
11651 means no Season defined.

11652 **Number of Week Profiles (mandatory):** Number of week profiles in the Week Profile Table. A
11653 value of 0x00 means no Week Profile defined.

11654 **Number of Day Profiles (mandatory):** Number of day profiles in the Day Profile Table.
11655

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

11656 D.9.2.3.2 **PublishDayProfile Command**

11657 The *PublishDayProfile* command is published in response to a *GetDayProfile* command. If the
11658 *IssuerCalendarID* does not match with one of the stored calendar instances, the client shall
11659 ignore the command and respond using ZCL Default Response with a status response of
11660 NOT_FOUND.

11661 The Calendar server shall send only the number of Schedule Entries belonging to this calendar
11662 instance. Server and clients shall be able to store at least 1 *DayProfile* for TOU and Auxiliary
11663 Load Switch calendars and three *DayProfiles* for a Friendly Credit calendar, and at least one
11664 *ScheduleEntries* per day profile. If the client is not able to store all *ScheduleEntries*, the device
11665 should respond using ZCL Default Response with a status response of INSUFFICIENT_SPACE.

11666 The ESI may send as many *PublishDayProfile* commands as needed, if the maximum application
11667 payload is not sufficient to transfer all *ScheduleEntries* in one command. In this case:

11668 • The *ScheduleEntries* shall be arranged in a linear array ordered by the start time.

11669 • The first command shall have *CommandIndex* set to 0, the second to 1 and so on.

11670 • The *Total Number of Commands* sub-field shall be set in all commands to the total number
11671 of **commands** being transferred.

11672 • The *Total Number of Schedule Entries* field shall be set in all commands to the total number
11673 of **entries** being transferred with the whole set of commands.

11674 • All associated commands shall use the same value of *Issuer Event ID*.

11675 D.9.2.3.2.1 Payload Format

Octets	**4**	**4**	**4**	**1**	**1**	**1**	**1**
Data Type	Unsigned 32-bit Integer	Unsigned 32-bit Integer	Unsigned 32-bit Integer	Unsigned 8-bit Integer	Unsigned 8-bit Integer	Unsigned 8-bit Integer	Unsigned 8-bit Integer
Field Name	Provider Id (M)	Issuer Event ID (M)	Issuer Calendar ID (M)	Day ID (M)	Total Number of Schedule Entries (M)	Command Index (M)	Total Number of Commands (M)

11676

Octets	**1**	**Variable**
Data Type	8-bit Enumeration	Series of Schedule Entries

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Field Name	Calendar Type (M)	Day Schedule Entries

Figure D-139– *PublishDayProfile* Command Payload

D.9.2.3.2.2 Payload Details

Provider Id (mandatory): An unsigned 32-bit field containing a unique identifier for the commodity provider. This field allows differentiation in deregulated markets where multiple commodity providers may be available.

Issuer Event ID (mandatory): Unique identifier generated by the commodity provider. When new information is provided that replaces older information for the same time period, this field allows devices to determine which information is newer. The value contained in this field is a unique number managed by upstream servers or a UTC based time stamp (UTCTime data type) identifying when the Publish command was issued. Thus, newer information will have a value in the Issuer Event ID field that is larger than older information.

Issuer Calendar ID (mandatory): Unique identifier generated by the commodity supplier. All parts of a calendar instance shall have the same *Issuer Calendar ID*.

Day ID (mandatory): Unique identifier generated by the commodity supplier. The *Day ID* is used as reference to assign a Day Profile to a Special Day or days in a Week Profile. When generating calendars, *Day IDs* shall be allocated sequentially, starting from 1.

Total Number of Schedule Entries (mandatory): An 8-bit integer representing the total number of *ScheduleEntries* in this Day Profile.

Command Index (mandatory): The *CommandIndex* is used to count the payload fragments in the case where the entire payload does not fit into one message. The *CommandIndex* starts at 0 and is incremented for each fragment belonging to the same command.

Total Number of Commands (mandatory): In the case where the entire payload does not fit into one message, the *Total Number of Commands* field indicates the total number of sub-commands in the message.

Calendar Type (mandatory): An 8-bit enumeration identifying the type of calendar published in this command. Table D-159 details the enumeration of this field. This field identifies the type of *Day Schedule Entry* included in this command.

D.9.2.3.2.3 Day Schedule Entries

The format of Day Schedule entries is dependent on the Calendar Type (see Table D-159). If the Calendar Type is 0x00 – 0x02 then Rate Start Times shall be used. If the value is 0x03 then the

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Friendly Credit Start Times shall be used. If the value is 0x04 then the Auxilliary Load Start Times shall be used. A value other than these would be invalid.

D.9.2.3.2.3.1 Schedule Entries for Rate Start Times

Schedule entries consist of a start time and the active price tier:

Octets	2	1
Data Type	Unsigned 16-bit Integer	8 bit Enumeration
Field Name	Start Time (M)	Price Tier (M)

Figure D-140– Schedule Entries for Rate Start Times Command Sub-Payload

Start Time (mandatory): The *Start Time* is represented in minutes from midnight. *ScheduleEntries* must be arranged in ascending order of *Start Times*. The first Schedule Entry must have 0x0000 (midnight) as the StartTime.

Price Tier (mandatory): This is the current price tier that is valid until the start time of the next Schedule Entry.

D.9.2.3.2.3.2 Schedule Entries for Friendly Credit Start Times

A *Friendly Credit Start Time* entry consists of a start time and an indication if Friendly Credit is available.

Octets	2	1
Data Type	Unsigned 16-bit Integer	Boolean
Field Name	Start Time (M)	Friendly Credit Enable (M)

Figure D-141– Schedule Entries for Friendly Credit Start Times Command Sub-Payload

Start Time (mandatory): The *Start Time* is represented in minutes from midnight. *ScheduleEntries* must be arranged in ascending order of *Start Times*. The first Schedule Entry must have 0x0000 (midnight) as the StartTime.

Friendly Credit Enable (mandatory): The *Friendly Credit Enable* field is a Boolean denoting if the Friendly Credit period is available for the consumer to use. A value of 1 means it is enabled and a 0 means that the Friendly Credit period is not available for the consumer to use.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

11727 ***D.9.2.3.2.3.3 Schedule Entries for Auxilliary Load Start Times***

11728 An *Auxilliary Load Start Time* entry consists of a start time, the relevant Auxiliary Switch and
11729 the state of the switch as a result of this action.

Octets	**2**	**1**
Data Type	Unsigned 16-bit Integer	8-bit BitMap
Field Name	Start Time (M)	Auxiliary Load Switch State (M)

11730 **Figure D-142– Schedule Entries for Auxilliary Load Start Times Command Sub-Payload**

11731 **Start Time (mandatory):** The *Start Time* is represented in minutes from midnight.
11732 *ScheduleEntries* must be arranged in ascending order of *Start Times*. The first Schedule Entry
11733 must have 0x0000 (midnight) as the StartTime.

11734 **Auxiliary Load Switch State (mandatory):** The required status of the auxiliary switches is
11735 indicated by the state of the bits. Bit0 correspond to Auxiliary Switch 1 and bit7 corresponds to
11736 Auxiliary Switch 8. A bit set to "1" indicates an ON state and a bit set to "0" indicates an OFF
11737 state.

11738 D.9.2.3.3 **PublishWeekProfile Command**

11739 The *PublishWeekProfile* command is published in response to a *GetWeekProfile* command. If the
11740 *IssuerCalendarID* does not match with one of the stored calendar instances, the client shall
11741 ignore the command and respond using ZCL Default Response with a status response of
11742 NOT_FOUND.

11743 The Calendar server shall send only the number of WeekProfiles belonging to this calendar
11744 instance. Server and clients shall be able to store at least 4 WeekProfiles for TOU calendars, and
11745 1 WeekProfile for Friendly Credit and Auxiliary Load Switch calendars. If the client is not able
11746 to store all entries, the device should respond using ZCL Default Response with a status response
11747 of INSUFFICIENT_SPACE.

11748 D.9.2.3.3.1 Payload Format

Octets	**4**	**4**	**4**	**1**
Data Type	Unsigned 32-bit Integer	Unsigned 32-bit Integer	Unsigned 32-bit Integer	Unsigned 8-bit Integer
Field Name	Provider Id (M)	Issuer Event ID (M)	Issuer Calendar ID (M)	Week ID (M)

11749

1	**1**	**1**	**1**	**1**	**1**	**1**

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Unsigned 8-bit Integer	Unsigned 8-bit Integer	Unsigned 8-bit Integer	Unsigned 8-bit Integer	Unsigned 8-bit Integer	Unsigned 8-bit Integer	Unsigned 8-bit Integer
Day ID Ref Monday	Day ID Ref Tuesday	Day ID Ref Wednesday	Day ID Ref Thursday	Day ID Ref Friday	Day ID Ref Saturday	Day ID Ref Sunday

11750 **Figure D-143– *PublishWeekProfile* Command Payload**

11751 D.9.2.3.3.2 Payload Details

11752 **Provider Id (mandatory):** An unsigned 32-bit field containing a unique identifier for the
11753 commodity provider. This field allows differentiation in deregulated markets where multiple
11754 commodity providers may be available.

11755 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. When
11756 new information is provided that replaces older information for the same time period, this field
11757 allows devices to determine which information is newer. The value contained in this field is a
11758 unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)
11759 identifying when the Publish command was issued. Thus, newer information will have a value in
11760 the *Issuer Event ID* field that is larger than older information.

11761 **Issuer Calendar ID (mandatory):** Unique identifier generated by the commodity supplier. All
11762 parts of a calendar instance shall have the same *Issuer Calendar ID*.

11763 **Week ID (mandatory):** Unique identifier generated by the commodity supplier. The *Week ID* is
11764 used as reference to assign a Week Profile to a Season Entry. When generating calendars, *Week*
11765 *IDs* shall be allocated sequentially, starting from 1.

11766 **Day ID Ref Monday** until **Day ID Ref Sunday (mandatory):** Reference to the related Day
11767 Profile entry.

11768 D.9.2.3.4 **PublishSeasons Command**

11769 The *PublishSeasons* command is published in response to a *GetSeason* command. If the
11770 *IssuerCalendarID* does not match with one of the stored calendar instances, the client shall
11771 ignore the command and respond using ZCL Default Response with a status response of
11772 NOT_FOUND.

11773 The Calendar server shall send only the number of *SeasonEntries* belonging to this calendar
11774 instance. Server and clients shall be able to store at least 4 SeasonEntries for TOU calendars, and
11775 1 SeasonEntry for Friendly Credit and Auxiliary Load Switch calendars. If the client is not able
11776 to store all *Season Entries*, the device should respond using ZCL Default Response with a status
11777 response of INSUFFICIENT_SPACE.

11778 The ESI may send as many *PublishSeasons* commands as needed, if the maximum application
11779 payload is not sufficient to transfer all Season Entries in one command. In this case:

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

- The *SeasonEntries* shall be arranged in a linear array ordered by the date.
- The first command shall have *Command Index* set to 0, the second to 1 and so on.
- The total number of seasons being transferred with the whole set of commands is known from the previously received *PublishCalendar* command.
- All associated commands shall use the same value of *Issuer Event ID*.

D.9.2.3.4.1 Payload Format

Octets	4	4	4	1	1	Variable
Data Type	Unsigned 32-bit Integer	Unsigned 32-bit Integer	Unsigned 32-bit Integer	Unsigned 8-bit Integer	Unsigned 8-bit Integer	Series of Season Entries
Field Name	Provider Id (M)	Issuer Event ID (M)	Issuer Calendar ID (M)	Command Index (M)	Total Number of Commands (M)	Season Entry

Figure D-144– *PublishSeasons* Command Payload

D.9.2.3.4.2 Payload Details

Provider Id (mandatory): An unsigned 32-bit field containing a unique identifier for the commodity provider. This field allows differentiation in deregulated markets where multiple commodity providers may be available.

Issuer Event ID (mandatory): Unique identifier generated by the commodity provider. When new information is provided that replaces older information for the same time period, this field allows devices to determine which information is newer. The value contained in this field is a unique number managed by upstream servers or a UTC based time stamp (UTCTime data type) identifying when the Publish command was issued. Thus, newer information will have a value in the *Issuer Event ID* field that is larger than older information.

Issuer Calendar ID (mandatory): Unique identifier generated by the commodity supplier. All parts of a calendar instance shall have the same *Issuer Calendar ID*.

Command Index (mandatory): The *Command Index* is used to count the payload fragments in the case where the entire payload does not fit into one message. The *Command Index* starts at 0 and is incremented for each fragment belonging to the same command.

Total Number of Commands (mandatory): In the case where the entire payload does not fit into one message, the *Total Number of Commands* field indicates the total number of sub-commands in the message.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

11805 **Season Entry:** A *Season Entry* consists of a *Season Start Date* and the reference (*Week ID Ref*)
11806 to the related Week Profile entry. The Start Date of the *Season Entries* must be arranged in
11807 ascending order. The active season is valid until the *Season Start Date* of the next *Season Entry*.

Octets	**4**	**1**
Data Type	Date	Unsigned 8-bit Integer
Field Name	Season Start Date (M)	Week ID Ref (M)

11808 **Figure D-145– Season Entry Sub-Payload**

11809 D.9.2.3.5 **PublishSpecialDays Command**

11810 The *PublishSpecialDays* command is published in response to a *GetSpecialDays* command or if
11811 a calendar update is available. If the *Calendar Type* does not match with one of the stored
11812 calendar instances, the client shall ignore the command and respond using ZCL Default
11813 Response with a status response of NOT_FOUND.

11814 The Calendar server shall send only the number of *SpecialDayEntries* belonging to this calendar
11815 instance. Server and clients shall be able to store at least 50 *SpecialDayEntries*. If the client is not
11816 able to store all *SpecialDayEntries*, the device should respond using ZCL Default Response with
11817 a status response of INSUFFICIENT_SPACE.

11818 If the maximum application payload is not sufficient to transfer all *SpecialDayEntries* in one
11819 command, the ESI may send as many *PublishSpecialDays* commands as needed. In this case:

11820 • The *SpecialDayEntries* shall be arranged in a linear array ordered by the date.

11821 • The first command shall have *Command Index* set to 0, the second to 1 and so on.

11822 • The *Total Number of SpecialDays* field shall be set in all commands to the total number of
11823 entries being transferred with the whole set of commands.

11824 • All associated commands shall use the same value of *Issuer Event ID*.

11825 Note that, in this case, it is the client's responsibility to ensure that it receives all associated
11826 *PublishSpecialDays* commands before any of the payloads can be used.

11827 D.9.2.3.5.1 Payload Format

11828 The *PublishSpecialDays* command shall be formatted as illustrated in Figure D-146:

Octets	**4**	**4**	**4**	**4**	**1**	**1**
Data Type	Unsigned 32-bit Integer	Unsigned 32-bit Integer	Unsigned 32-bit Integer	UTC Time	8-bit Enumeration	Unsigned 8-bit Integer

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Field Name	Provider Id (M)	Issuer Event ID (M)	Issuer Calendar Id (M)	Start Time (M)	Calendar Type (M)	Total Number of SpecialDays (M)

11829

Octets	1	1	Variable
Data Type	Unsigned 8-bit Integer	Unsigned 8-bit Integer	Series of Special Days
Field Name	Command Index (M)	Total Number of Commands (M)	Special Day Entry

11830 **Figure D-146– *PublishSpecialDays* Command Payload**

11831 D.9.2.3.5.2 Payload Details

11832 **Provider Id (mandatory):** An unsigned 32-bit field containing a unique identifier for the
11833 commodity provider. This field allows differentiation in deregulated markets where multiple
11834 commodity providers may be available.

11835 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. When
11836 new information is provided that replaces older information for the same time period, this field
11837 allows devices to determine which information is newer. The value contained in this field is a
11838 unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)
11839 identifying when the Publish command was issued. Thus, newer information will have a value in
11840 the Issuer Event ID field that is larger than older information. If multiple *PublishSpecialDays*
11841 commands are needed to transfer the whole Special Day Table, the commands belonging to the
11842 same Special Day Table shall use the same *IssuerEventID* and *StartTime*.

11843 **Issuer Calendar ID (mandatory):** Unique identifier generated by the commodity Supplier. All
11844 parts of a calendar instance shall have the same *Issuer Calendar ID*.

11845 **Start Time (mandatory):** A UTC Time field to denote the time at which the Special Day Table
11846 becomes valid. A start date/time of 0x00000000 shall indicate that the command should be
11847 executed immediately. A start date/time of 0xFFFFFFFF shall cause an existing
11848 *PublishSpecialDays* command with the same *Provider ID* and *Issuer Event ID* to be cancelled
11849 (note that, in markets where permanently active price information is required for billing
11850 purposes, it is recommended that a replacement/superseding *PublishSpecialDays* command is
11851 used in place of this cancellation mechanism).

11852 **Calendar Type (mandatory):** An 8-bit enumeration identifying the type of calendar this day
11853 profile belongs to. Generation Meters shall use the 'Received' Calendar. See Table D-159.

11854 **Total Number of SpecialDays (mandatory):** An 8-bit integer representing the total number of
11855 Special Day entries in this Special Day Table.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Command Index (mandatory): The *Command Index* is used to count the payload fragments in the case where the entire payload does not fit into one message. The *Command Index* starts at 0 and is incremented for each fragment belonging to the same command.

Total Number of Commands (mandatory): In the case where the entire payload does not fit into one message, the *Total Number of Commands* field indicates the total number of sub-commands in the message.

SpecialDayEntry: A *SpecialDayEntry* consists of the *Special Day Date* and a reference (*Day ID Ref*) to the related Day Profile entry. The dates of the Special Day Table must be arranged in ascending order.

Octets	**4**	**1**
Data Type	Date	Unsigned 8-bit Integer
Field Name	Special Day Date (M)	Day ID Ref (M)

Figure D-147– *SpecialDayEntry* Sub-Payload

D.9.2.3.6 **CancelCalendar Command**

The *CancelCalendar* command indicates that all data associated with a particular calendar instance should be discarded.

In markets where permanently active price (and hence calendar) information is required for billing purposes, it is recommended that replacement/superseding *PublishCalendar*, *PublishDayProfile*, *PublishWeekProfile* and *PublishSeasons* commands are used in place of a *CancelCalendar* command. The exception is a 'Friendly Credit' calendar, where an instance is not always required.

D.9.2.3.6.1 Payload Format

The *CancelCalendar* command shall be formatted as illustrated in Figure D-148:

Octets	**4**	**4**	**1**
Data Type	Unsigned 32-bit Integer	Unsigned 32-bit Integer	8-bit Enumeration
Field Name	Provider Id (M)	Issuer Calendar Id (M)	Calendar Type (M)

Figure D-148– *CancelCalendar* Command Payload

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

11878 D.9.2.3.6.2 Payload Details

11879 **Provider Id (mandatory):** An unsigned 32-bit field containing a unique identifier for the
11880 commodity provider. This field allows differentiation in deregulated markets where multiple
11881 commodity providers may be available.

11882 **Issuer Calendar ID (mandatory):** Unique identifier generated by the commodity Supplier. All
11883 parts of a calendar instance shall have the same *Issuer Calendar ID*.

11884 **Calendar Type (mandatory):** An 8-bit enumeration identifying the type of calendar to be
11885 cancelled by this command. Table D-159 details the enumeration of this field.

11886 D.9.2.3.6.3 Effect on Receipt

11887 On receipt of this command, a client device shall discard all instances of *PublishCalendar*,
11888 *PublishDayProfile*, *PublishWeekProfile*, *PublishSeasons* and *PublishSpecialDays* commands
11889 associated with the stated *Provider ID, Calendar Type* and *Issuer Calendar ID*.

11890

11891 ### D.9.2.4 Commands Received

11892 Table D-161 lists cluster-specific commands that are received by the server.

11893 **Table D-161– Commands Received by the Calendar Cluster Server**

Command Identifier Field Value	Description	Mandatory / Optional
0x00	GetCalendar	O
0x01	GetDayProfiles	O
0x02	GetWeekProfiles	O
0x03	GetSeasons	O
0x04	GetSpecialDays	O
0x05	GetCalendarCancellation	O

11894

11895 D.9.2.4.1 **GetCalendar Command**

11896 This command initiates *PublishCalendar* command(s) for scheduled Calendar updates. To obtain
11897 the complete Calendar details, further *GetDayProfiles*, *GetWeekProfiles* and *GetSeasons*
11898 commands must be sent using the *IssuerCalendarID* obtained from the appropriate
11899 *PublishCalendar* command.

11900 D.9.2.4.1.1 Payload Format

Octets	**4**	**4**	**1**	**1**	**4**

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Data Type	UTC Time	Unsigned 32-bit Integer	Unsigned 8-bit Integer	8-bit Enumeration	Unsigned 32-bit Integer
Field Name	Earliest Start Time (M)	Min. Issuer Event ID (M)	Number of Calendars (M)	Calendar Type (M)	Provider Id (M)

Figure D-149– GetCalendar Command Payload

D.9.2.4.1.2 Payload Details

Earliest Start Time (mandatory): UTC Timestamp indicating the earliest start time of calendars to be returned by the corresponding *PublishCalendar* command. The first returned *PublishCalendar* command shall be the instance which is active or becomes active at or after the stated *Earliest Start Time*. If more than one instance is requested, the active and scheduled instances shall be sent with ascending ordered *Start Time*.

Min. Issuer Event ID (mandatory): A 32-bit integer representing the minimum *Issuer Event ID* of calendars to be returned by the corresponding *PublishCalendar* command. A value of 0xFFFFFFFF means not specified; the server shall return calendars irrespective of the value of the *Issuer Event ID*.

Number of Calendars (mandatory): An 8-bit integer which represents the maximum number of *PublishCalendar* commands that the client is willing to receive in response to this command. A value of 0 would indicate all available *PublishCalendar* commands shall be returned.

Calendar Type (mandatory): An 8-bit enumeration identifying the calendar type of the requested calendar. Generation Meters shall use the 'Received' Calendar. See Table D-159. A value of 0xFF means not specified. If the *CalendarType* is not specified, the server shall return calendars regardless of its type.

Provider Id (mandatory): An unsigned 32-bit field containing a unique identifier for the commodity provider. This field allows differentiation in deregulated markets where multiple commodity providers may be available. A value of 0xFFFFFFFF means not specified; the server shall return calendars irrespective of the value of the *Provider Id*.

D.9.2.4.2 **GetDayProfiles Command**

This command initiates one or more *PublishDayProfile* commands for the referenced Calendar.

D.9.2.4.2.1 Payload Format

Octets	4	4	1	1
Data Type	Unsigned 32-bit Integer	Unsigned 32-bit Integer	Unsigned 8-bit Integer	Unsigned 8-bit Integer

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Field Name	Provider Id (M)	Issuer Calendar ID (M)	Start Day Id (M)	Number of Days (M)

Figure D-150– GetDayProfiles Command Payload

D.9.2.4.2.2 Payload Details

Provider Id (mandatory): An unsigned 32-bit field containing a unique identifier for the commodity provider. This field allows differentiation in deregulated markets where multiple commodity providers may be available. A value of 0xFFFFFFFF means not specified; the server shall return day profiles irrespective of the value of the *Provider Id*.

Issuer Calendar ID (mandatory): *IssuerCalendarID* of the calendar to which the requested Day Profiles belong.

Start Day ID (mandatory): Unique identifier for a Day Profile generated by the commodity supplier. The *Start Day ID* indicates the minimum ID of Day Profiles to be returned by the corresponding *PublishDayProfile* command. A value of 0x01 indicates that the (first) *PublishDayProfile* command should contain the profile with the lowest Day ID held by the server. A value of 0x00 is unused.

Number of Days (mandatory): An 8-bit integer which represents the maximum number of Day Profiles that the client is willing to receive in response to this command. A value of 0x00 will cause the return of all day profiles with an ID equal to or greater than the *Start Day ID*.

Note**:** A Day Profile table may need multiple *PublishDayProfile* commands to be transmitted to the client.

D.9.2.4.3 **GetWeekProfiles Command**

This command initiates one or more *PublishWeekProfile* commands for the referenced Calendar.

D.9.2.4.3.1 Payload Format

Octets	4	4	1	1
Data Type	Unsigned 32-bit Integer	Unsigned 32-bit Integer	Unsigned 8-bit Integer	Unsigned 8-bit Integer
Field Name	Provider Id (M)	Issuer Calendar ID (M)	Start Week Id (M)	Number of Weeks (M)

Figure D-151– GetWeekProfiles Command Payload

D.9.2.4.3.2 Payload Details

Provider Id (mandatory): An unsigned 32-bit field containing a unique identifier for the commodity provider. This field allows differentiation in deregulated markets where multiple

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

commodity providers may be available. A value of 0xFFFFFFFF means not specified; the server shall return week profiles irrespective of the value of the *Provider Id*.

Issuer Calendar ID (mandatory): *IssuerCalendarID* of the calendar to which the requested Week Profiles belong.

Start Week ID (mandatory): Unique identifier for a Week Profile generated by the commodity supplier. The *Start Week ID* indicates the minimum ID of Week Profiles to be returned by the corresponding *PublishWeekProfile* command. A value of 0x01 indicates that the *PublishWeekProfile* command should contain the profile with the lowest Week ID held by the server. A value of 0x00 is unused.

Number of Weeks (mandatory): An 8-bit integer which represents the maximum number of Week Profiles that the client is willing to receive in response to this command. A value of 0x00 will cause the return of all week profiles with an ID equal to or greater than the *Start Week ID*.

D.9.2.4.4 GetSeasons Command

This command initiates one or more *PublishSeasons* commands for the referenced Calendar.

D.9.2.4.4.1 Payload Format

Octets	**4**	**4**
Data Type	Unsigned 32-bit Integer	Unsigned 32-bit Integer
Field Name	Provider Id (M)	Issuer Calendar ID (M)

Figure D-152– GetSeasons Command Payload

Provider Id (mandatory): An unsigned 32-bit field containing a unique identifier for the commodity provider. This field allows differentiation in deregulated markets where multiple commodity providers may be available. A value of 0xFFFFFFFF means not specified; the server shall return season tables irrespective of the value of the *Provider Id*.

Issuer Calendar ID (mandatory): *IssuerCalendarID* of the calendar to which the requested Seasons belong.

Note: A Season Table may need multiple *PublishSeasons* commands to be transmitted to the client.

D.9.2.4.5 GetSpecialDays Command

This command initiates one or more *PublishSpecialDays* commands for the scheduled Special Day Table updates.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

11979 D.9.2.4.5.1 Payload Format

Octets	4	1	1	4	4
Data Type	UTC Time	Unsigned 8-bit Integer	8-bit Enumeration	Unsigned 32-bit Integer	Unsigned 32-bit Integer
Field Name	Start Time (M)	Number of Events (M)	Calendar Type (M)	Provider Id (M)	Issuer Calendar ID (M)

11980 **Figure D-153– GetSpecialDays Command Payload**

11981 D.9.2.4.5.2 Payload Details

11982 **Start Time (mandatory):** UTC Timestamp to select active and scheduled events to be returned
11983 by the corresponding *PublishSpecialDays* command. If the command has a *Start Time* of
11984 0x00000000, replace that *Start Time* with the current time stamp.

11985 **Number of Events (mandatory):** An 8-bit integer which represents the maximum number of
11986 Special Day Table instances to be sent. A value of 0 would indicate all available Special Day
11987 tables shall be returned. The first returned *PublishSpecialDays* command should be that which is
11988 active or becomes active at the stated *Start Time*. The first returned Special Day table shall be the
11989 instance which is active or becomes active at the stated *Start Time*. If more than one instance is
11990 requested, the active and scheduled instances shall be sent with ascending ordered *Start Time*.

11991 Note: A Special Day table may need multiple *PublishSpecialDay* commands to be transmitted to
11992 the client.

11993 **Calendar Type (mandatory):** An 8-bit enumeration identifying the calendar type of the
11994 requested Special Days. Generation Meters shall use the 'Received' Calendar. See Table D-159.
11995 A value of 0xFF means not specified. If the *CalendarType* is not specified, the server shall return
11996 Special Days regardless of their type.

11997 **Provider Id (mandatory):** An unsigned 32-bit field containing a unique identifier for the
11998 commodity provider. This field allows differentiation in deregulated markets where multiple
11999 commodity providers may be available. A value of 0xFFFFFFFF means not specified; the server
12000 shall return Special Day tables irrespective of the value of the *Provider Id*.

12001 **Issuer Calendar ID (mandatory):** Unique identifier generated by the commodity supplier. A
12002 value of 0x00000000 will cause the return of all Special Days profiles.
12003

12004 D.9.2.4.6 **GetCalendarCancellation Command**

12005 This command initiates the return of the last *CancelCalendar* command held on the associated
12006 server.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

12007 D.9.2.4.6.1 Payload Details

12008 This command has no payload.

12009 D.9.2.4.6.2 When Generated

12010 This command is generated when the client device wishes to fetch any pending *CancelCalendar*
12011 command from the server (see D.9.2.3.6 for further details). In the case of a BOMD, this may be
12012 as a result of the associated Notification flag.

12013 A ZCL Default response with status NOT_FOUND shall be returned if there is no
12014 *CancelCalendar* command available.

12015

12016 ## D.9.3 Client

12017 ### D.9.3.1 Dependencies

12018 Support for ZCL Data Types.

12019 No dependencies exist for other Smart Energy clusters.

12020 ### D.9.3.2 Attributes

12021 The client has no attributes.

12022 ### D.9.3.3 Commands Received

12023 The client receives the cluster-specific response commands detailed in D.9.2.3.

12024 ### D.9.3.4 Commands Generated

12025 The client generates the cluster-specific commands detailed in D.9.2.4, as required by the
12026 application.

12027

12028 ## D.9.4 Application Guidelines

12029 The following notes should be read in conjunction with the overview in section D.9.1.

12030 It is recommended that mains-powered client devices ‘register’ (bind) with an associated
12031 Calendar server in order to receive new calendar information as soon as it becomes available.
12032 Calendar servers should publish new calendar information to bound clients as soon as it is
12033 successfully received by the server.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Battery-powered devices, or device with limited resources, should not bind to the Calendar cluster. These devices are expected to poll the Calendar server regularly in order to check for updates to calendar items.

It is recommended that calendar information is persisted on devices throughout a reboot or power-cycle. However, ALL devices should request the latest calendar information following power up, after a reboot, or following any period without HAN communication.

Acquisition of a calendar starts when a client asks for or gets pushed a current or pending *PublishCalendar* command. From the information contained in the *PublishCalendar* command, the client should request the relevant day, week and/or season information, respectively utilizing *GetDayProfiles*, *GetWeekProfiles* and *GetSeasons* commands.

There may be specific days when special switching behavior overrides the normal one defined by the Season or Week Profile tables. These special dates are contained within a Special Day Table associated with the particular calendar instance. As Special Day Table information may change more frequently than the other information contained within a calendar, any update to the Special Day Table will be sent unsolicited to Calendar clients registered with the relevant Calendar server. Battery-powered devices are expected to poll the Calendar server regularly for updates to the Special Day information in a similar way to that used for other calendar information.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

D.10 Device Management Cluster

D.10.1 Overview

The Device Management Cluster provides an interface to the functionality of devices within a Smart energy Network. The cluster will support the following functions:

- Supplier Control
- Tenancy Control
- Password Control
- Event Configuration

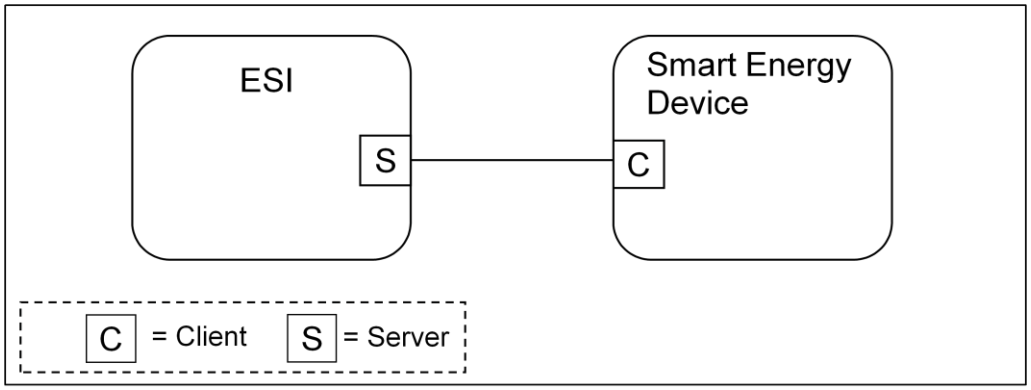


Figure D-154– Device Management Cluster Client/Server Example

D.10.1.1 Supplier Control

This functionality provides a method to control the activities required to change the energy supplier to the premises (CoS).

D.10.1.2 Tenancy Control

This functionality provides a method to control the activities required when changing the tenant (consumer) of the property (CoT).

D.10.1.3 Password Control

Passwords or PINs are used to protect access to consumer data or to secure access to the energy supplier's meter service menus.

The Password commands provide a mechanism where a specific password located on a Smart Energy device may be changed to a new value or reset. The server shall maintain an access

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

12077 control list of the type of password required vs. the device and, where applicable, store the last
12078 password for the device. Each device that supports this feature shall have a local default
12079 password.

12080 The server shall send unsolicited *RequestNewPasswordResponse* commands to its clients (except
12081 BOMDs unless unsolicited messages are enabled in its policy) when the backhaul connection
12082 requires the device to update the password.
12083

12084 ## D.10.2 Server

12085 ### D.10.2.1 Dependencies

12086 Events carried using this cluster include a timestamp with the assumption that target devices
12087 maintain a real time clock. Devices can acquire and synchronize their internal clocks via the ZCL
12088 Time server.

12089

12090 ### D.10.2.2 Attributes

12091 For convenience, the attributes defined in this specification are arranged into sets of related
12092 attributes; each set can contain up to 256 attributes. Attribute identifiers are encoded such that the
12093 most significant Octet specifies the attribute set and the least significant Octet specifies the
12094 attribute within the set. The currently defined attribute sets are listed in the following Table
12095 D-162.

12096 **Table D-162– Device Management Attribute Sets**

Attribute Set Identifier	Description
0x00	Reserved
0x01	Supplier Control Attribute Set
0x02	Tenancy Control Attribute Set
0x03	Backhaul Control Attribute Set
0x04	HAN Control Attribute Set
0x05 – 0xFF	Reserved

12097

12098 D.10.2.2.1 **Supplier Control Attribute Set**

12099 **Table D-163– Supplier Control Attribute Set**

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x0100	ProviderID	Unsigned 32bit Integer	0x00000000 – 0xFFFFFFFF	Read only	0x00000000	O
0x0101	ProviderName	Octet String	1 - 17	Read only	-	O
0x0102	ProviderContactDetails	Octet String	1 - 20	Read only	-	O
0x0103 – 0x010F	Reserved					
0x0110	ProposedProviderID	Unsigned 32bit Integer	0x00000000 – 0xFFFFFFFF	Read only	-	O
0x0111	ProposedProviderName	Octet String	1 - 17	Read only	-	O
0x0112	ProposedProvider ChangeDate/Time	UTC Time	0x00000000 – 0xFFFFFFFF	Read only	-	O
0x0113	ProposedProvider ChangeControl	32-Bit Bitmap	0x00000000 – 0xFFFFFFFF	Read only	-	O
0x0114	ProposedProvider ContactDetails	Octet String	1 - 20	Read only	-	O
0x0115 – 0x011F	Reserved					
0x0120	ReceivedProviderID	Unsigned 32bit Integer	0x00000000 – 0xFFFFFFFF	Read only	-	O
0x0121	ReceivedProviderName	Octet String	1 - 17	Read only	-	O
0x0122	ReceivedProvider ContactDetails	Octet String	1 - 20	Read only	-	O
0x0123 – 0x012F	Reserved					
0x0130	ReceivedProposed ProviderID	Unsigned 32bit Integer	0x00000000 – 0xFFFFFFFF	Read only	-	O
0x0131	ReceivedProposed Provider Name	Octet String	1 - 17	Read only	-	O
0x0132	ReceivedProposed Provider ChangeDate/Time	UTC Time	0x00000000 – 0xFFFFFFFF	Read only	-	O
0x0133	ReceivedProposed Provider ChangeControl	32-Bit Bitmap	0x00000000 – 0xFFFFFFFF	Read only	-	O
0x0134	ReceivedProposed Provider ContactDetails	Octet String	1 - 20	Read only	-	O
0x0135 – 0x01FF	Reserved					

12100

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

12101 D.10.2.2.1.1 Provider ID Attribute

12102 An unsigned 32-bit field containing a unique identifier for the current commodity supplier. The
12103 default value of 0x00000000 shall be used for installation.

12104 D.10.2.2.1.2 Provider Name Attribute

12105 An octet string containing the name of the current supplier of the commodity to the device. The
12106 attribute is capable of storing a 16 character string (the first octet indicates length) encoded in the
12107 UTF-8 format.

12108 D.10.2.2.1.3 Provider Contact Details Attribute

12109 An octet string containing the contact details of the current Provider delivering a commodity to
12110 the premises. The attribute is capable of storing a 19 character string (the first octet indicates
12111 length) encoded in UTF-8 format.

12112 D.10.2.2.1.4 Proposed Provider ID Attribute

12113 An unsigned 32-bit field containing a unique identifier for the commodity supplier associated
12114 with the proposed change to the supply of the commodity.

12115 D.10.2.2.1.5 Proposed Provider Name Attribute

12116 The *Proposed Provider Name* indicates the name for the commodity supplier associated with the
12117 proposed change to the supply of energy. This attribute is an octet string field capable of storing
12118 a 16 character string (the first octet indicates length) encoded in the UTF-8 format.

12119 D.10.2.2.1.6 Proposed Provider Change Date/Time Attribute

12120 A UTC time that defines the time and date when the new supplier will take over the supply of the
12121 commodity to the Meter/HAN.

12122 D.10.2.2.1.7 Proposed Provider Change Control

12123 This is a 32-bit mask that denotes the functions that are required to be carried out on processing
12124 of the change of supplier. The format of this Bitmap is shown within Table D-164.

12125 **Table D-164– Proposed Change Control**

Bit	Value	Description
0	Pre Snapshots (see Metering & Prepayment clusters for additional information)	A snapshot shall be triggered
1	Post Snapshots (see Metering & Prepayment clusters for additional information)	A snapshot shall be triggered
2	Reset Credit Register	All Credit Registers shall be reset to their default value
3	Reset Debit Register	All Debt Registers shall be reset to their default value

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

4	Reset Billing Period	All Billing periods shall be reset to their default value
5	Clear Tariff Plan	The tariff shall be reset to its default value
6	Clear Standing Charge	The Standing Charge shall be reset to its default value
7	Block Historical Load Profile Information	Historical LP information shall no longer be available to be published to the HAN. With regards to a meter that is mirrored, this information may be available to the HES but not to the HAN. Any historical LP shall be cleared from the IHD.
8	Clear Historical Load Profile Information	Historical LP information shall be cleared from all devices
9	Clear IHD Data - Consumer	All consumer data shall be removed
10	Clear IHD Data - Supplier	All supplier data shall be removed
11 & 12	Meter Contactor State "On / Off / Armed"	The required status of the meter contactor post action. Available bit combinations are shown in Table D-165. NOTE: In certain markets, this value cannot trigger automatic reconnection of the supply, only maintain the current status of, disconnect or ARM the supply.
13	Clear Transaction Log	All transaction logs shall be cleared from all devices
14	Clear Prepayment Data	All Prepayment Registers shall be reset to their default state
15 – 31	Reserved	

Table D-165– Contactor State Bit Combinations

Bit Combination	Status
0b00	Supply OFF
0b01	Supply OFF / ARMED
0b10	Supply ON (see note)
0b11	Supply UNCHANGED

D.10.2.2.1.8 Proposed Provider Contact Details Attribute

An octet string containing the contact details of the Provider associated with the proposed change of supply of the commodity delivered to the premises. The attribute is capable of storing a 19 character string (the first octet indicates length) encoded in UTF-8 format.

D.10.2.2.1.9 ReceivedProviderID Attribute

An unsigned 32-bit field containing a unique identifier for the commodity supplier receiving the Received energy.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

12136 D.10.2.2.1.10 ReceivedProviderName Attribute

12137 The name of the current supplier of Received energy services to the device. This attribute is an
12138 octet string field capable of storing a 16 character string (the first octet indicates length) encoded
12139 in the UTF-8 format.

12140 D.10.2.2.1.11 ReceivedProviderContactDetails Attribute

12141 An octet string containing the contact details of the current Provider receiving a commodity from
12142 the premises. The attribute is capable of storing a 19 character string (the first octet indicates
12143 length) encoded in UTF-8 format.

12144 D.10.2.2.1.12 ReceivedProposedProviderID Attribute

12145 An unsigned 32-bit field containing a unique identifier for the commodity supplier associated
12146 with the proposed change to the Receiving of energy.

12147 D.10.2.2.1.13 ReceivedProposedProviderName Attribute

12148 The *Received Proposed Provider Name* indicates the name for the commodity supplier
12149 associated with the proposed change to the Receiving of energy. This attribute is an octet string
12150 field capable of storing a 16 character string (the first octet indicates length) encoded in the UTF-
12151 8 format.

12152 D.10.2.2.1.14 ReceivedProposedProviderChangeDate/Time Attribute

12153 A UTC time that defines the time and date that the new supplier will take over the Received of
12154 energy from the Meter/HAN.

12155 D.10.2.2.1.15 ReceivedProposedProviderChangeControl Attribute

12156 This is a 32-bit mask that denotes the functions that are required to be carried out on processing
12157 of the change of supplier. The format of this Bitmap is shown within Table D-164.

12158 D.10.2.2.1.16 Received Proposed Provider Contact Details Attribute

12159 An octet string containing the contact details of the Provider associated with the proposed change
12160 of receipt of the commodity from the premises. The attribute is capable of storing a 19 character
12161 string (the first octet indicates length) encoded in UTF-8 format.

12162 D.10.2.2.2 **Tenancy Control Attribute Set**

12163 **Table D-166– Tenancy Control Attribute Set**

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0200	ChangeofTenancy UpdateDate/Time	UTC Time		Read only	-	O
0x0201	Proposed Tenancy Change Control	32-Bit Bitmap	0x00000000 – 0xFFFFFFFF	Read only	-	O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x0202 – 0x02FF	Reserved					

12164

12165 D.10.2.2.2.1 ChangeofTenancyUpdateDate/Time Attribute

12166 The *ChangeofTenancyUpdateDate/Time* attribute indicates the time at which a proposed change
12167 to the tenancy is to be implemented. Until an initial change of tenancy becomes available, this
12168 attribute shall be set to 0xFFFFFFFF (i.e. invalid).

12169 D.10.2.2.2.2 ProposedTenancyChangeControl Attribute

12170 This is a 32-bit mask that denotes the functions that are required to be carried out on processing
12171 of the change of tenancy. The format of this Bitmap is shown within Table D-164. Until an initial
12172 change of tenancy becomes available, this attribute shall be set to 0x00000000.

12173 D.10.2.2.3 **Backhaul Control Attribute Set**

12174 **Table D-167– Backhaul Control Attribute Set**

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0300	WAN Status	8 bit Enumeration	0x00 – 0xFF	Read only	-	O
0x0301 – 0x03FF	Reserved					

12175

12176 D.10.2.2.3.1 WAN Status Attribute

12177 The *WAN Status* attribute is an 8-bit enumeration defining the state of the WAN (Wide Area
12178 Network) connection as listed in the table below:

12179 **Table D-168– State of the WAN Connection**

Enumeration	Description
0x00	Connection to WAN is not available
0x01	Connection to WAN is available
0x02 - 0xFF	Reserved

12180

12181 D.10.2.2.4 **HAN Control Attribute Set**

12182 **Table D-169– HAN Control Attribute Set**

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0400	LowMediumThreshold	Unsigned 32-bit Integer	0x00000000 – 0xFFFFFFFF	Read Only		O
0x0401	MediumHighThreshold	Unsigned 32-bit Integer	0x00000000 – 0xFFFFFFFF	Read Only		O
0x0402 – 0x04FF	Reserved					

12183

12184 D.10.2.2.4.1 Low Medium Threshold Attribute

12185 The *Low Medium Threshold* attribute is an unsigned 32-bit integer indicating the threshold at
12186 which the value of *Instantaneous Demand* is deemed to have moved from low energy usage to
12187 medium usage. The unit of measure for this value is as specified by the *UnitOfMeasure* attribute
12188 within the Metering cluster (see Table D-26 for definition).

12189 D.10.2.2.4.2 Medium High Threshold Attribute

12190 The *Medium High Threshold* attribute is an unsigned 32-bit integer indicating the threshold at
12191 which the value of *Instantaneous Demand* is deemed to have moved from medium energy usage
12192 to high usage. The unit of measure for this value is as specified by the *UnitOfMeasure* attribute
12193 within the Metering cluster (see Table D-26 for definition).
12194

12195 ### D.10.2.3 Commands Received

12196 Table D-170 lists the cluster specific commands that are received by the server.

12197 **Table D-170– Cluster Specific Commands Received by the Server**

Command Identifier Field Value	Description	Mandatory / Optional
0x00	Get Change of Tenancy	O
0x01	Get Change of Supplier	O
0x02	Request New Password	O
0x03	GetSiteID	O
0x04	Report Event Configuration	O
0x05	GetCIN	O
0x06– 0xFF	Reserved	

12198

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

12199 D.10.2.3.1 **Get Change of Tenancy Command**

12200 This command is used to request the ESI to respond with information regarding any available
12201 change of tenancy.

12202 D.10.2.3.1.1 Payload Details

12203 There are no fields for this command.

12204 D.10.2.3.1.2 Effect on Receipt

12205 The ESI shall send a *PublishChangeofTenancy* command. A ZCL Default Response with status
12206 NOT_FOUND shall be returned if there is no change of tenancy information available.

12207

12208 D.10.2.3.2 **Get Change of Supplier Command**

12209 This command is used to request the ESI to respond with information regarding any available
12210 change of supplier.

12211 D.10.2.3.2.1 Payload Details

12212 There are no fields for this command.

12213 D.10.2.3.2.2 Effect on Receipt

12214 The ESI shall send a *PublishChangeofSupplier* command. A ZCL Default Response with status
12215 NOT_FOUND shall be returned if there is no change of supplier information available.

12216

12217 D.10.2.3.3 **RequestNewPassword Command**

12218 This command is used to request the current Password from the server.

12219 D.10.2.3.3.1 Payload Format

Octets	**1**
Data Type	Unsigned 8-bit Enumeration
Field Name	Password Type (M)

12220 **Figure D-155– RequestNewPassword Command Payload**

12221 D.10.2.3.3.2 Payload Details

12222 **PasswordType (mandatory):** Indicates which password is requested. The possible password
12223 types are defined in Table D-172.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

12224 D.10.2.3.3.3 Effect on Receipt

12225 The ESI shall send a *RequestNewPasswordResponse* command. A ZCL Default Response with
12226 status NOT_FOUND shall be returned if there is no password available.

12227

12228 D.10.2.3.4 **GetSiteID Command**

12229 This command is used to request the ESI to respond with information regarding any pending
12230 change of Site ID.

12231 D.10.2.3.4.1 Payload Details

12232 There are no fields for this command.

12233 D.10.2.3.4.2 Effect on Receipt

12234 The ESI shall send an *UpdateSiteID* command. A ZCL Default Response with status
12235 NOT_FOUND shall be returned if there is no change of Site ID pending.

12236

12237 D.10.2.3.5 **Report Event Configuration Command**

12238 This command is sent in response to a *GetEventConfiguration* command.

12239 D.10.2.3.5.1 Payload Format

Octets	**1**	**1**	**variable**
Data Type	Unsigned 8-bit Integer	Unsigned 8-bit Integer	…
Field Name	Command Index (M)	Total Commands (M)	Event Configuration Payload (M)

12240 **Figure D-156– Report Event Configuration Command Payload**

12241 D.10.2.3.5.2 Payload Details

12242 **Command Index (mandatory):** The *Command Index* is used to count the payload fragments in
12243 the case where the entire payload does not fit into one message. The *Command Index* starts at 0
12244 and is incremented for each fragment belonging to the same command.

12245 **Total Commands (mandatory):** This parameter holds the total number of responses.

12246 **Event Configuration Payload (mandatory):** The log payload is a series of events, in time
12247 sequential order. The event payload consists of the logged events and detailed within the event
12248 configuration attribute list:

Octets	**2**	**1**	**…**	**2**	**1**

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Data Type	Unsigned 16-bit Integer	8-bit Bitmap	…	Unsigned 16-bit Integer	8-bit Bitmap
Field Name	Event ID (M)	Event Configuration (M)	…	Event ID n (M)	Event Configuration n (M)

12249 **Figure D-157– Report Event Configuration Sub-Payload**

12250 **Event ID (mandatory):** The *Event ID* is the attribute ID of the Event Configuration attribute.
12251 ZigBee Event IDs are detailed in Table D-176 to Table D-184.

12252 **Event Configuration (mandatory)**: The configuration bitmap applicable to the event, as defined
12253 in Table D-177.

12254

12255 D.10.2.3.6 **GetCIN Command**

12256 This command is used to request the ESI to respond with information regarding any pending
12257 change of Customer ID Number.

12258 D.10.2.3.6.1 Payload Details

12259 There are no fields for this command.

12260 D.10.2.3.6.2 Effect on Receipt

12261 The ESI shall send an *UpdateCIN* command. A ZCL Default Response with status
12262 NOT_FOUND shall be returned if there is no change of Customer ID Number pending.
12263
12264

12265 ## D.10.2.4 Commands Generated

12266 Table D-171 lists the cluster specific commands that are generated by the server.

12267 **Table D-171– Cluster Specific Commands Generated by the Server**

Command Identifier	Description	Mandatory / Optional
0x00	Publish Change of Tenancy	O
0x01	Publish Change of Supplier	O
0x02	Request New password Response	O
0x03	UpdateSiteID	O
0x04	SetEventConfiguration	O
0x05	GetEventConfiguration	O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x06	UpdateCIN	O
0x07 – 0xFF	Reserved	

12268 D.10.2.4.1 **Publish Change of Tenancy Command**

12269 This command is used to change the tenancy of a meter.

12270 D.10.2.4.1.1 Payload Format

Octets	**4**	**4**	**1**	**4**	**4**
Data Type	Unsigned 32 bit Integer	Unsigned 32 bit Integer	8 bit Bitmap	UTC Time	32 bit BitMap
Field Name	Provider ID (M)	Issuer Event ID (M)	Tariff Type (M)	Implementation Date/Time(M)	Proposed Tenancy Change Control (M)

12271 **Figure D-158– Publish Change of Tenancy Command Payload**

12272 D.10.2.4.1.2 Payload Details

12273 **Provider ID (mandatory):** An unsigned 32 bit field containing a unique identifier for the
12274 commodity provider to whom this command relates.

12275 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. When
12276 new information is provided that replaces older information for the same time period, this field
12277 allows devices to determine which information is newer. The value contained in this field is a
12278 unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)
12279 identifying when the Publish command was issued. Thus, newer information will have a value in
12280 the *Issuer Event ID* field that is greater than older information.

12281 **Tariff Type (Mandatory):** An 8-bit bitmap identifying the type of tariff published in this
12282 command. The least significant nibble represents an enumeration of the tariff type as detailed in
12283 Table D-108 (Generation Meters shall use the 'Received' Tariff). The most significant nibble is
12284 reserved.

12285 **Implementation Date/Time (mandatory):** A UTC Time field to indicate the date from which
12286 the change of tenancy is to be applied. This value shall always be in advance of the
12287 *CommandDate/Time* and/or the *LocalTime* by at least 24hrs. An *Implementation Date/Time* of
12288 0xFFFFFFFF shall cause an existing but pending *Publish Change of Tenancy* command with the
12289 same *Provider ID* and *Issuer Event ID* to be cancelled.

12290 **Proposed Tenancy Change Control (mandatory):** A 32-bit mask that denotes the functions
12291 that are required to be carried out on processing of this command. See Table D-164 for further
12292 details.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

12293 D.10.2.4.1.3 When Generated

12294 The *PublishChangeofTenancy* command shall be generated from the ESI, and sent to the meter,
12295 when a change of tenancy is required. This command can be sent prior to the change of tenancy.
12296 The meter should use the standard ZCL response.

12297 D.10.2.4.1.4 Effect on Receipt

12298 On receipt of the *PublishChangeofTenancy* command, the device shall update the
12299 *ChangeofTenancyUpdateDate/Time* and *ProposedTenancyChangeControl* attributes, but only
12300 action the command at the *ImplementationDate/Time.* At the *ImplementationDate/Time,* the
12301 device shall check the *ProposedTenancyChangeControl* attribute to understand what additional
12302 action(s) it must carry out pre and post the change.

12303 D.10.2.4.2 **Publish Change of Supplier Command**

12304 This command is used to change the Supplier (commodity provider) that is supplying the
12305 property. This command shall only be used if there is a requirement for the *ProviderID* to be a
12306 static value within the Prepayment and Price clusters. Should there be a requirement for the
12307 *ProviderID* to be dynamic, this command and the associated attributes should not be used. It is
12308 recommended that this command is sent at least one week before the proposed date of change.

12309 D.10.2.4.2.1 Payload Format

Octets	**4**	**4**	**1**	**4**	**4**	**4**	**1 - 16**	**1 - 20**
Data Type	Unsigned 32 bit Integer	Unsigned 32 bit Integer	8 bit Bitmap	Unsigned 32-bit Integer	UTC Time	32-Bit BitMap	Octet String	Octet String
Field Name	Current Provider ID (M)	Issuer Event ID (M)	Tariff Type (M)	Proposed Provider ID (M)	Provider Change Implementation Time (M)	Provider Change Control (M)	Proposed Provider Name (M)	Proposed Provider Contact Details (M)

12310 **Figure D-159– Publish Change of Supplier Command Payload**

12311 D.10.2.4.2.2 Payload Details

12312 **Current Provider ID (mandatory):** An unsigned 32 bit field containing a unique identifier for
12313 the current commodity provider to whom this command relates.

12314 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. When
12315 new information is provided that replaces older information for the same time period, this field
12316 allows devices to determine which information is newer. The value contained in this field is a
12317 unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)
12318 identifying when the Publish command was issued. Thus, newer information will have a value in
12319 the *Issuer Event ID* field that is larger than older information.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Tariff Type (Mandatory): An 8-bit bitmap identifying the type of tariff published in this command. The least significant nibble represents an enumeration of the tariff type as detailed in Table D-108 (Generation Meters shall use the 'Received' Tariff). The most significant nibble is reserved.

Proposed Provider ID (mandatory): An unsigned 32 bit field containing a unique identifier for the commodity provider associated with the proposed change to the supply. Depending on the *Tariff Type*, this value will be taken from either attribute D.10.2.2.1.4 or D.10.2.2.1.12.

Provider Change Implementation Time (mandatory): A UTC Time field to indicate the date/time at which a proposed change to the provider is to be implemented. Depending on the *Tariff Type*, this value will be taken from either attribute D.10.2.2.1.6 or D.10.2.2.1.14. A *Provider Change Implementation Time* of 0xFFFFFFFF shall cause an existing but pending *Publish Change of Supplier* command with the same Current *Provider ID* and *Issuer Event ID* to be cancelled.

Proposed Provider Name (mandatory): An octet string that denotes the name of the new commodity provider. This is dependent on the *Tariff Type* value; for Received, the parameter should match the attribute in section D.10.2.2.1.13, for all other values it should match the attribute in section D.10.2.2.1.5.

Proposed Provider Contact Details (mandatory): An octet string that denotes the contact details of the new commodity provider. The field shall be capable of storing a 19 character string (the first octet indicates length) encoded in UTF-8 format.

Provider Change Control (mandatory): A 32-bit mask that denotes the functions that are required to be carried out on processing of this command. See section D.10.2.2.1.7 or D.10.2.2.1.15, depending on the *Tariff Type*.

D.10.2.4.2.3 When Generated

The *PublishChangeofSupplier* command shall be generated from the ESI, and sent to the meter, when a change of commodity provider is required. It shall also be generated in response to a *Get Change of Supplier* command. The *PublishChangeofSupplier* command contains a start date/time which allows the command to be sent in advance of the changeover date.

D.10.2.4.2.4 Effect on Receipt

Following receipt of a *PublishChangeofSupplier* command, the meter shall only action the command at the *ProviderChangeImplementationTime.* At this point in time, the meter shall check the *Provider Change Control* field to understand what action(s) it must carry out pre and post the change.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

12354 D.10.2.4.3 **Request New Password Response Command**

12355 This command is used to send the current password to the client. A
12356 *RequestNewPasswordResponse* command is sent either as a response to a *RequestNewPassword*
12357 command or unsolicited when the HES has changed the password.

12358 D.10.2.4.3.1 Payload Format

Octets	**4**	**4**	**2**	**1**	**11**
Data Type	Unsigned 32-bit Integer	UTC Time	Unsigned 16-bit Integer	Unsigned 8-bit Enumeration	Octet String
Field Name	Issuer Event ID (M)	Implementation Date/Time (M)	Duration in minutes (M)	Password Type(M)	Password (M)

12359 **Figure D-160– RequestNewPasswordResponse Command Payload**

12360 D.10.2.4.3.2 Payload Details

12361 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. When
12362 new information is provided that replaces older information for the same time period, this field
12363 allows devices to determine which information is newer. The value contained in this field is a
12364 unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)
12365 identifying when the Publish command was issued. Thus, newer information will have a value in
12366 the *Issuer Event ID* field that is larger than older information.

12367 **Implementation Date/Time (mandatory):** A UTC Time field to indicate the date at which the
12368 originating command was to be applied.

12369 **Duration in minutes (mandatory):** An unsigned 16-bit integer that denotes the duration in
12370 minutes that the password is valid for. A value of Zero means the password is valid until
12371 changed.

12372 **PasswordType (mandatory):** Indicates which password should be changed. The possible
12373 password types are defined in Table D-172. The password types can be used flexibly by various
12374 end devices. The scope of authority assigned to a password type should be defined by the
12375 corresponding end device.

12376 **Table D-172– Password Type Enumeration**

Enumerated Value	**Description**	**Usage**
0x00	Reserved	Not Used
0x01	Password 1	Used for access to the Service menu
0x02	Password 2	Used for access to the Consumer menu
0x03	Password 3	TBD
0x04	Password 4	TBD
0x05 to 0xFF	Reserved	

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Password (mandatory): An octet string of length 11 that contains the password (the first octet is the length, allowing 10 octets for the password).

D.10.2.4.3.3 Effect on Receipt

On receipt of this command, the client shall update the specified password.

D.10.2.4.4 **Update SiteID Command**

This command is used to set the *SiteID* attribute on a meter (see D.3.2.2.4.8).

D.10.2.4.4.1 Payload Format

Octets	**4**	**4**	**4**	**1-33**
Data Type	Unsigned 32 bit Integer	UTC Time	Unsigned 32 bit Integer	Octet String
Field Name	Issuer Event ID (M)	SiteID Time (M)	Provider ID (M)	SiteID (M)

Figure D-161– Update SiteID Command Payload

D.10.2.4.4.2 Payload Details

Issuer Event ID (mandatory): Unique identifier generated by the commodity provider. When new information is provided that replaces older information for the same time period, this field allows devices to determine which information is newer. The value contained in this field is a unique number managed by upstream servers or a UTC based time stamp (UTCTime data type) identifying when the Publish command was issued. Thus, newer information will have a value in the *Issuer Event ID* field that is larger than older information.

SiteID Time (mandatory): A UTC Time field to denote the time at which the update of *SiteID* will take place. A date/time of 0x00000000 shall indicate that the command should be executed immediately (comparison against a time source should NOT be made in this case). A date/time of 0xFFFFFFFF shall cause an existing but pending *Update SiteID* command with the same *Provider ID* and *Issuer Event ID* to be cancelled.

Provider ID: An unsigned 32-bit field containing a unique identifier for the commodity provider to whom this command relates.

SiteID (mandatory): An octet string that denotes the Site ID.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

12403 D.10.2.4.5 **SetEventConfiguration Command**

12404 This command provides a method to set the event configuration attributes, held in a client device.

12405 D.10.2.4.5.1 Payload Format

Octets	**4**	**4**	**1**	**1**	**Variable**
Data Type	Unsigned 32–bit Integer	UTC Time	8-bit Bitmap	8-bit Enumeration	…
Field Name	Issuer Event ID (M)	Start Date/Time (M)	Event Configuration (M)	Configuration Control (M)	Event Configuration Payload (M)

12406 **Figure D-162– Set Event Configuration Command Payload**

12407 D.10.2.4.5.2 Payload Details

12408 **Issuer Event ID (mandatory)**: Unique identifier generated by the commodity provider. When
12409 new information is provided that replaces older information for the same time period, this field
12410 allows devices to determine which information is newer. The value contained in this field is a
12411 unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)
12412 identifying when the command was issued. Thus, newer information will have a value in the
12413 *Issuer Event ID* field that is larger than older information.

12414 **Start Date/Time (mandatory):** A UTC Time field to indicate the date and time at which the
12415 new configuration is to be applied.

12416 **Event Configuration (mandatory):** This field holds the new event configuration to be applied,
12417 as defined in Table D-177.

12418 **Configuration Control (mandatory):** The *Configuration Control* enumeration allows the new
12419 configuration value to be applied to several events via a single command. The value of this field
12420 defines the format of the event configuration payload:

12421 **Table D-173– Configuration Control Enumeration**

Value	**Description**
0x00	Apply by List
0x01	Apply by Event Group
0x02	Apply by Log Type
0x03	Apply by Configuration Match
0x04 – 0FF	Reserved

12422

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

12423 ***D.10.2.4.5.2.1 Apply by List***

12424 The ‘Apply by List’ option allows individual or lists of events to be configured by a single
12425 command:

Octets	1	2	…	2
Data Type	Unsigned 8 bit Integer	Unsigned 16-bit Integer	…	Unsigned 16-bit Integer
Field Name	Number of Events (M)	Event ID 1 (M)	…	Event ID n (M)

12426 **Figure D-163– ‘Apply by List’ Sub-Payload**

12427 **Number of Events (mandatory)**: This field holds the number of events contained within the
12428 command.

12429 **Event ID (mandatory)**: The *Event ID* is the attribute ID of the event configuration attribute.
12430 ZigBee Event IDs are detailed in Table D-176 to Table D-184.

12431 ***D.10.2.4.5.2.2 Apply by Event Group***

12432 The ‘Apply by Event Group’ option allows all events belonging to a stated event group (attribute
12433 set) to be configured by a single command:

Octets	2
Data Type	Unsigned 16 bit Integer
Field Name	Event Group ID (M)

12434 **Figure D-164– ‘Apply by Event Group’ Sub-Payload**

12435 **Event Group ID (mandatory)**: The *Event Group ID* field indicates which attribute set the event
12436 belongs to (see Table D-174). The *Event Group ID* is in the form ‘0xnnFF’, where *nn* is the
12437 Attribute Set Identifier (the final attribute in the sets defined in Table D-176 to Table D-184 is
12438 reserved as a ‘wildcard’ attribute to allow definition of the *Event Group IDs*.

12439 ***D.10.2.4.5.2.3 Apply by Log Type***

12440 The ‘Apply by Log Type’ option allows all configurations recorded in a given log to be
12441 configured:

Octets	1
Data Type	Unsigned 8 bit Integer
Field Name	Log ID

12442 **Figure D-165– ‘Apply by Log Type’ Sub-Payload**

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

12443 **Log ID**: The *Log ID* specifies the log ID of events to be updated with the new *Configuration*
12444 *Value* field passed in the command. The applicable values for this field are defined by bits 0-2 of
12445 the Table D-177.

12446 ***D.10.2.4.5.2.4 Apply by Configuration Match***

12447 The 'Apply by Configuration Match' option allows all events matching a given configuration
12448 value to be changed to the new configuration value:

Octets	**1**
Data Type	8-bit Bitmap
Field Name	Configuration Value Match (M)

12449 **Figure D-166– 'Apply by Configuration Match' Sub-Payload**

12450 **Configuration Value Match (mandatory)**: This field indicates that any configuration attribute
12451 which matches this value shall be assigned the new configuration value passed in the *Event*
12452 *Configuration* field of the main command payload (see D.10.2.4.5.1).

12453 D.10.2.4.6 **GetEventConfiguration Command**

12454 This command allows the server to request details of event configurations.

12455 D.10.2.4.6.1 Payload Format

Octets	**2**
Data Type	Unsigned 16-bit Integer
Field Name	Event ID (M)

12456 **Figure D-167– Get Event Configuration Command**

12457 D.10.2.4.6.2 Payload Details

12458 **Event ID (mandatory):** The *Event ID* specifies a particular event to be queried. A value of
12459 0xFFFF is reserved to indicate all event IDs. A value equal to the *Event Group ID (*the final
12460 attribute in the sets defined in Table D-176 to Table D-184 is reserved for this purpose) shall
12461 indicate all event IDs within the indicated attribute set. The ZigBee Event IDs are detailed in
12462 Table D-176 to Table D-184.

12463

12464 D.10.2.4.7 **Update CIN Command**

12465 This command is used to set the *CustomerIDNumber* attribute held in the Metering cluster (see
12466 D.3.2.2.4.18).

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

12467 D.10.2.4.7.1 Payload Format

Octets	4	4	4	1-25
Data Type	Unsigned 32 bit Integer	UTC Time	Unsigned 32 bit Integer	Octet String
Field Name	Issuer Event ID (M)	CIN Implementation Time (M)	Provider ID (M)	CustomerID Number (M)

12468 **Figure D-168– Update CIN Command Payload**

12469 D.10.2.4.7.2 Payload Details

12470 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. When
12471 new information is provided that replaces older information for the same time period, this field
12472 allows devices to determine which information is newer. The value contained in this field is a
12473 unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)
12474 identifying when the Publish command was issued. Thus, newer information will have a value in
12475 the *Issuer Event ID* field that is larger than older information.

12476 **CIN Implementation Time (mandatory):** A UTC Time field to denote the date/time at which
12477 the updated *CustomerIDNumber* will become active. A value of 0x00000000 shall indicate that
12478 the command should be executed immediately (comparison against a time source should NOT be
12479 made in this case). A value of 0xFFFFFFFF shall cause an existing but pending *UpdateCIN*
12480 command with the same *Provider ID* and *Issuer Event ID* to be cancelled.

12481 **Provider ID (mandatory):** An unsigned 32-bit field containing a unique identifier for the
12482 commodity provider to whom this command relates.

12483 **CustomerIDNumber (mandatory):** An octet string that denotes the Customer ID Number.

12484

12485 D.10.2.4.7.3 Effect on Receipt

12486 Upon successful receipt of this command, the meter shall update the *CustomerIDNumber*
12487 attribute and return a ZCL Default Response indicating SUCCESS.

12488 A ZCL Default Response indicating NOT_AUTHORIZED shall be returned if the Provider ID
12489 contained within the command does not match the current Provider ID. For all other failures, a
12490 ZCL Default Response indicating FAILURE shall be returned.

12491

12492

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

12493 ## D.10.3 Client

12494 ### D.10.3.1 Dependencies

12495 • Support for ZCL Data Types

12496 • Events carried using this cluster include a timestamp with the assumption that target devices
12497 maintain a real time clock. Devices can acquire and synchronize their internal clocks via the
12498 ZCL Time server.

12499 ### D.10.3.2 Attributes

12500 **Table D-174– Device Management Client Attribute Sets**

Attribute Set Identifier	Description
0x00	Supplier Attribute Set
0x01	Price Event Configuration Attribute Set
0x02	Metering Event Configuration Attribute Set
0x03	Messaging Event Configuration Attribute set
0x04	Prepay Event Configuration Attribute Set
0x05	Calendar Event Configuration Attribute set
0x06	Device Management Event Confguration Attribute set
0x07	Tunnel Event Configuration Attribute set
0x08	OTA Event Configuration Attribute set
0x09 – 0x7F	Reserved
0x80 – 0xFF	Reserved for non-ZigBee Event Configuration

12501

12502 #### D.10.3.2.1 **Supplier Attribute Set**

12503 **Table D-175– Supplier Attribute Sets**

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0000	Provider ID	Unsigned 32bit Integer	0x00000000 – 0xFFFFFFFF	Read only	-	O
0x0001 – 0x000F	Reserved					
0x0010	ReceivedProvider ID	Unsigned 32bit Integer	0x00000000 – 0xFFFFFFFF	Read only	-	O
0x0011 – 0x00FF	Reserved					

12504

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

12505 D.10.3.2.1.1 ProviderID Attribute

12506 An unsigned 32 bit field containing a unique identifier for the commodity provider to whom this
12507 attribute relates.

12508 D.10.3.2.1.2 ReceivedProviderID Attribute

12509 An unsigned 32 bit field containing a unique identifier for the commodity provider to whom this
12510 attribute relates. This attribute is only for the Received supply.

12511

12512 D.10.3.2.2 **Price Event Configuration Attribute Set**

12513 The following attributes allow events related to pricing to be configured.

12514 It should be noted that triggers for events are an implementation issue, however it is suggested
12515 that the 'Tariff Activated' events should only be logged (if configured to do so) when moving
12516 from one tariff type to another, not when a tariff is modified.

12517 **Table D-176– Price Event Configuration Attribute Set**

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory/ Optional
0x0100	TOUTariffActivation	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0101	BlockTariffactivated	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0102	BlockTOUTariffActivated	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0103	SingleTariffRateActivated	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0104	AsychronousBillingOccurred	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0105	SynchronousBillingOccurred	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0106	Tariff NotSupported	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0107	PriceClusterNotFound	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0108	CurrencyChangePassiveActivated	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0109	CurrencyChangePassiveUpdated	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x010A	PriceMatrixPassiveActivated	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x010B	PriceMatrixPassiveUpdated	8-bit Bitmap	0x00 – 0xFF	Read only		O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory/ Optional
0x010C	TariffChangePassiveActivated	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x010D	TariffChangedPassiveUpdated	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x010E - 0x01AF	RESERVED					
0x01B0	PublishPriceReceived	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01B1	PublishPriceActioned	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01B2	PublishPriceCancelled	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01B3	PublishPriceRejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01B4	PublishTariffInformation Received	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01B5	PublishTariffInformation Actioned	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01B6	PublishTariffInformation Cancelled	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01B7	PublishTariffInformation Rejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01B8	PublishPriceMatrixReceived	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01B9	PublishPriceMatrixActioned	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01BA	PublishPriceMatrixCancelled	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01BB	PublishPriceMatrixRejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01BC	PublishBlockThresholdsReceived	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01BD	PublishBlockThresholdsActioned	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01BE	PublishBlockThresholdsCancelled	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01BF	PublishBlockThresholdsRejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01C0	PublishCalorificValueReceived	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01C1	PublishCalorificValueActioned	8-bit Bitmap	0x00 – 0xFF	Read only		O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory/ Optional
0x01C2	PublishCalorificValueCancelled	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01C3	PublishCalorificValueRejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01C4	PublishConversionFactorReceived	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01C5	PublishConversionFactorActioned	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01C6	PublishConversionFactorCancelled	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01C7	PublishConversionFactorRejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01C8	PublishCO_2ValueReceived	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01C9	PublishCO_2ValueActioned	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01CA	PublishCO_2ValueCancelled	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01CB	PublishCO_2ValueRejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01CC	PublishCPPEventReceived	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01CD	PublishCPPEventActioned	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01CE	PublishCPPEventCancelled	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01CF	PublishCPPEventRejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01D0	PublishTierLabelsReceived	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01D1	PublishTierLabelsActioned	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01D2	PublishTierLabelsCancelled	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01D3	PublishTierLabelsRejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01D4	PublishBillingPeriodReceived	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01D5	PublishBillingPeriodActioned	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01D6	PublishBillingPeriodCancelled	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01D7	PublishBillingPeriodRejected	8-bit Bitmap	0x00 – 0xFF	Read only		O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory/ Optional
0x01D8	PublishConsolidatedBillReceived	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01D9	PublishConsolidatedBillActioned	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01DA	PublishConsolidatedBillCancelled	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01DB	PublishConsolidatedBillRejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01DC	PublishBlockPeriodReceived	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01DD	PublishBlockPeriodActioned	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01DE	PublishBlockPeriodCancelled	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01DF	PublishBlockPeriodRejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01E0	PublishCreditPaymentInfoReceived	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01E1	PublishCreditPaymentInfoActioned	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01E2	PublishCreditPaymentInfoCancelled	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01E3	PublishCreditPaymentInfoRejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01E4	PublishCurrencyConversionReceived	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01E5	PublishCurrencyConversionActioned	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01E6	PublishCurrencyConversionCancelled	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01E7	PublishCurrencyConversionRejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01E8– 0x01FE	RESERVED					
0x01FF	Reserved for Price cluster Group ID	--	--	Read only	-	O

12518

12519 D.10.3.2.2.1 Event Configuration Attributes

12520 The least-significant 3 bits of the Event Configuration bitmaps indicate how the event should be
12521 logged; the remaining bits provide options for treatment rules to be applied.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

12522 **Table D-177– Event Configuration Bitmaps**

Bit	Description	
Bits 0-2	**Enumerated Value**	**Description**
	0	Do not Log
	1	Log as Tamper
	2	Log as Fault
	3	Log as General Event
	4	Log as Security Event
	5	Log as Network Event
	6-7	Reserved
Bit 3	Push Event to WAN	
Bit 4	Push Event to HAN	
Bit 5	Raise Alarm (ZigBee)	
Bit 6	Raise Alarm (Physical i.e. audible/visible)	
Bit 7	Reserved	

12523

12524 D.10.3.2.3 **Metering Event Configuration Attribute Set**

12525 The following attributes allow events related to the meter to be configured.

12526 **Table D-178– Metering Event Configuration Attribute Set**

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0200	Check Meter	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0201	Low Battery	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0202	Tamper Detect	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0203	Supply Status	8-bit Bitmap	0x00 – 0xFF	Read only		O
	Electricity: Power Failure					
	Gas: Not Defined					
	Water: Pipe Empty					
	Heat/Cooling: Temperature Sensor					
0x0204	Supply Quality	8-bit Bitmap	0x00 – 0xFF	Read only		O
	Electricity: Power Quality					
	Gas: Low Pressure					

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
	Water: Low Pressure *Heat/Cooling: Burst Detect*					
0x0205	Leak Detect	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0206	Service Disconnect	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0207	Reverse Flow *Electricity: Reserved* *Gas: Reverse Flow* *Water: Reverse Flow* *Heat/Cooling: Flow Sensor*	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0208	MeterCoverRemoved	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0209	MeterCoverClosed	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x020A	Strong MagneticField	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x020B	NoStrongMagneticField	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x020C	BatteryFailure	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x020D	ProgramMemoryError	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x020E	RAMError	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x020F	NVMemoryError	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0210	LowVoltageL1	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0211	HighVoltageL1	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0212	LowVoltageL2	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0213	HighVoltageL2	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0214	LowVoltageL3	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0215	HighVoltageL3	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0216	OverCurrentL1	8-bit Bitmap	0x00 – 0xFF	Read only		O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0217	OverCurrentL2	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0218	OverCurrentL3	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0219	FrequencyTooLowL1	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x021A	FrequencyTooHighL1	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x021B	FrequencyTooLowL2	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x021C	FrequencyTooHighL2	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x021D	FrequencyTooLowL3	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x021E	FrequencyTooHighL3	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x021F	GroundFault	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0220	ElectricTamperDetect	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0221	IncorrectPolarity	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0222	CurrentNoVoltage	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0223	UnderVoltage	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0224	OverVoltage	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0225	NormalVoltage	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0226	PFBelowThreshold	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0227	PFAboveThreshold	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0228	TerminalCoverRemoved	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0229	TerminalCoverClosed	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x022A – 0x022F	RESERVED					
0x0230	BurstDetect	8-bit Bitmap	0x00 – 0xFF	Read only		O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0231	PressureTooLow	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0232	PressureTooHigh	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0233	FlowSensorCommunicationError	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0234	FlowSensorMeasurementFault	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0235	FlowSensorReverseFlow	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0236	Flow sensor air detect	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0237	PipeEmpty	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0238 – 0x023F	RESERVED					
0x0240 – 0x024F	RESERVED (Water Specific Alarm Group)					
0x0250	InletTemperatureSensorFault	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0251	OutletTemperatureSendorFault	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0252 – 0x025F	RESERVED					
0x0260	ReverseFlow	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0261	TiltTamper	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0262	BatteryCoverRemoved	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0263	BatteryCoverClosed	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0264	ExcessFlow	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0265	Tilt Tamper Ended	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0266 – 0x026F	RESERVED					
0x0270	MeasurementSystemError	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0271	WatchdogError	8-bit Bitmap	0x00 – 0xFF	Read only		O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0272	SupplyDisconnectFailure	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0273	SupplyConnectFailure	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0274	MeasurementSoftwareChanged	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0275	DSTenabled	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0276	DSTdisabled	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0277	ClockAdjBackward	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0278	ClockAdjForward	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0279	ClockInvalid	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x027A	CommunicationErrorHAN	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x027B	CommunicationOKHAN	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x027C	MeterFraudAttempt	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x027D	PowerLoss	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x027E	UnusualHANTraffic	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x027F	UnexpectedClockChange	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0280	CommsUsingUnauthenticated Component	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0281	ErrorRegClear	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0282	AlarmRegClear	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0283	UnexpectedHWReset	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0284	UnexpectedProgramExecution	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0285	LimitThresholdExceeded	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0286	LimitThresholdOK	8-bit Bitmap	0x00 – 0xFF	Read only		O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0287	LimitThresholdChanged	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0288	MaximumDemandExceeded	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0289	ProfileCleared	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x028A	LoadProfileCleared	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x028B	BatteryWarning	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x028C	WrongSignature	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x028D	NoSignature	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x028E	SignatureNotValid	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x028F	UnauthorisedActionfromHAN	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0290	FastPollingStart	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0291	FastPollingEnd	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0292	MeterReportingInterval Changed	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0293	DisconnecttoLoadLimit	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0294	MeterSupplyStatusRegister Changed	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0295	MeterAlarmStatusRegister Changed	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0296	ExtendedMeterAlarmStatus Register Changed.	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0297	DataAccessViaLocalPort	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0298	Configure Mirror Success	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0299	Configure Mirror Failure	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x029A	Configure Notification Flag Scheme Success	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x029B	Configure Notification Flag Scheme Failure	8-bit Bitmap	0x00 – 0xFF	Read only		O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x029C	Configure Notification Flags Success	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x029D	Configure Notification Flags Failure	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x029E	Stay Awake Request HAN	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x029F	Stay Awake Request WAN	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02A0 – 0x02AF	RESERVED					
0x02B0	ManufacturerSpecificA	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02B1	ManufacturerSpecificB	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02B2	ManufacturerSpecificC	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02B3	ManufacturerSpecificD	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02B4	ManufacturerSpecificE	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02B5	ManufacturerSpecificF	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02B6	ManufacturerSpecificG	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02B7	ManufacturerSpecificH	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02B8	ManufacturerSpecificI	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02B9 – 0x02BF	RESERVED					
0x02C0	Get Profile Command Received	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02C1	Get Profile Command Actioned	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02C2	Get Profile Command Cancelled	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02C3	Get Profile Command Rejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02C4	RequestMirrorResponse Command Received	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02C5	RequestMirrorResponse Command Actioned	8-bit Bitmap	0x00 – 0xFF	Read only		O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x02C6	RequestMirrorResponse Command Cancelled	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02C7	RequestMirrorResponse Command Rejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02C8	MirrorRemoved Command Received	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02C9	MirrorRemoved Command Actioned	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02CA	MirrorRemoved Command Cancelled	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02CB	MirrorRemoved Command Rejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02CC	GetSnapshot Command Received	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02CD	GetSnapshot Command Actioned	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02CE	GetSnapshot Command Cancelled	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02CF	GetSnapshot Command Rejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02D0	TakeSnapshot Command Received	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02D1	TakeSnapshot Command Actioned	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02D2	TakeSnapshot Command Cancelled	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02D3	TakeSnapshot Command Rejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02D4	MirrorReportAttributeResponse Command Received	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02D5	MirrorReportAttributeResponse Command Actioned	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02D6	MirrorReportAttributeResponse Command Cancelled	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02D7	MirrorReportAttributeResponse Command Rejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02D8	ScheduleSnapshot Command Received	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02D9	ScheduleSnapshot Command Actioned	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02DA	ScheduleSnapshot Command Cancelled	8-bit Bitmap	0x00 – 0xFF	Read only		O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x02DB	ScheduleSnapshot Command Rejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02DC	StartSampling Command Received	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02DD	StartSampling Command Actioned	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02DE	StartSampling Command Cancelled	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02DF	StartSampling Command Rejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02E0	GetSampledData Command Received	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02E1	GetSampledData Command Actioned	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02E2	GetSampledData Command Cancelled	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02E3	GetSampledData Command Rejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02E4	Supply ON	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02E5	Supply ARMED	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02E6	Supply OFF	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02E7	Disconnected due to Tamper Detected.	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02E8	ManualDisconnect	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02E9	ManualConnect	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02EA	RemoteDisconnection	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02EB	RemoteConnect	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02EC	LocalDisconnection	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02ED	LocalConnect	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02EE	Change Supply Received	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02EF	Change Supply Actioned	8-bit Bitmap	0x00 – 0xFF	Read only		O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x02F0	Change Supply Cancelled	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02F1	Change Supply Rejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02F2	Local Change Supply Received	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02F3	Local Change Supply Actioned	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02F4	Local Change Supply Cancelled	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02F5	Local Change Supply Rejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02F6	PublishUncontrolledFlow Threshold Received	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02F7	PublishUncontrolledFlow Threshold Actioned	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02F8	PublishUncontrolledFlow Threshold Cancelled	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02F9	PublishUncontrolledFlow Threshold Rejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x02FA-0x02FE	RESERVED					
0x02FF	Reserved for Metering cluster Group ID	--	--	Read only	-	O

12527

12528 D.10.3.2.3.1 Event Configuration Attributes

12529 The attributes in this set allow a server device to configure how an event is handled when
12530 triggered. All attributes in this set are bitmaps as defined in Table D-177.

12531 D.10.3.2.4 **Messaging Event Configuration Attribute Set**

12532 The following attributes allow events related to messaging to be configured.

12533 **Table D-179– Messaging Event Configuration Attribute Set**

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0300	Message Confirmation Sent	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0301 – 0x03BF	RESERVED					

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x03C0	DisplayMessageReceived	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x03C1	DisplayMessageActioned	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x03C2	DisplayMessageCancelled	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x03C3	DisplayMessageRejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x03C4	CancelMessageReceived	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x03C5	CancelMessageActioned	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x03C6	CancelMessageCancelled	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x03C7	CancelMessageRejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x03C8 – 0x03FE	RESERVED					
0x03FF	Reserved for Messaging cluster Group ID	--	--	Read only	-	O

12534

12535 D.10.3.2.4.1 Event Configuration Attributes

12536 The attributes in this set allow a server device to configure how an event is handled when
12537 triggered. All attributes in this set are bitmaps as defined in Table D-177.

12538 D.10.3.2.5 **Prepayment Event Configuration Attribute Set**

12539 The following attributes allow events related to prepayment to be configured.

12540 **Table D-180– Prepayment Event Configuration Attribute Set**

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0400	Low Credit	8-bit Bitmap	0x00 – 0xFF	Read only	-	O
0x0401	No Credit (Zero Credit)	8-bit Bitmap	0x00 – 0xFF	Read only	-	O
0x0402	Credit Exhausted	8-bit Bitmap	0x00 – 0xFF	Read only	-	O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0403	Emergency Credit Enabled	8-bit Bitmap	0x00 – 0xFF	Read only	-	O
0x0404	Emergency Credit Exhausted	8-bit Bitmap	0x00 – 0xFF	Read only	-	O
0x0405	IHD Low Credit Warning	8-bit Bitmap	0x00 – 0xFF	Read only	-	O
0x0406 – 0x041F	RESERVED					
0x0420	Physical Attack on the Prepay Meter	8-bit Bitmap	0x00 – 0xFF	Read only	-	O
0x0421	Electronic Attack on the Prepay Meter	8-bit Bitmap	0x00 – 0xFF	Read only	-	O
0x0422	Discount Applied	8-bit Bitmap	0x00 – 0xFF	Read only	-	O
0x0423	Credit Adjustment	8-bit Bitmap	0x00 – 0xFF	Read only	-	O
0x0424	Credit Adjust Fail	8-bit Bitmap	0x00 – 0xFF	Read only	-	O
0x0425	Debt Adjustment	8-bit Bitmap	0x00 – 0xFF	Read only	-	O
0x0426	Debt Adjust Fail	8-bit Bitmap	0x00 – 0xFF	Read only	-	O
0x0427	Mode Change	8-bit Bitmap	0x00 – 0xFF	Read only	-	O
0x0428	Topup Code Error	8-bit Bitmap	0x00 – 0xFF	Read only	-	O
0x0429	Topup Already Used	8-bit Bitmap	0x00 – 0xFF	Read only	-	O
0x042A	Topup Code Invalid	8-bit Bitmap	0x00 – 0xFF	Read only	-	O
0x042B	Topup Accepted via Remote	8-bit Bitmap	0x00 – 0xFF	Read only	-	O
0x042C	Topup Accepted via Manual Entry	8-bit Bitmap	0x00 – 0xFF	Read only	-	O
0x042D	Friendly Credit In Use	8-bit Bitmap	0x00 – 0xFF	Read only	-	O
0x042E	Friendly Credit Period End Warning	8-bit Bitmap	0x00 – 0xFF	Read only	-	O
0x042F	Friendly Credit Period End	8-bit Bitmap	0x00 – 0xFF	Read only	-	O
0x0430	ErrorRegClear	8-bit Bitmap	0x00 – 0xFF	Read only	-	O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0431	AlarmRegClear	8-bit Bitmap	0x00 – 0xFF	Read only	-	O
0x0432	Prepay Cluster Not Found	8-bit Bitmap	0x00 – 0xFF	Read only	-	O
0x0433	Topup Value Too Large	8-bit Bitmap	0x00 – 0xFF	Read only	-	O
0x0434-0x0440	RESERVED					
0x0441	ModeCredit2Prepay	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0442	ModePrepay2Credit	8-bit Bitmap	0x00 – 0xFF	Read only	-	O
0x0443	ModeDefault	8-bit Bitmap	0x00 – 0xFF	Read only	-	O
0x0444 – 0x04BF	RESERVED					
0x04C0	SelectAvailableEmergencyCredit Received	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x04C1	SelectAvailableEmergencyCredit Actioned	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x04C2	SelectAvailableEmergencyCredit Cancelled	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x04C3	SelectAvailableEmergencyCredit Rejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x04C4	Change Debt Received	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x04C5	Change Debt Actioned	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x04C6	Change Debt Cancelled	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x04C7	Change Debt Rejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x04C8	Emergency Credit Setup Received	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x04C9	Emergency Credit Setup Actioned	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x04CA	Emergency Credit Setup Cancelled	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x04CB	Emergency Credit Setup Rejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x04CC	Consumer Topup Received	8-bit Bitmap	0x00 – 0xFF	Read only		O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x04CD	Consumer Topup Actioned	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x04CE	Consumer Topup Cancelled	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x04CF	Consumer Topup Rejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x04D0	Credit Adjustment Received	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x04D1	Credit Adjustment Actioned	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x04D2	Credit Adjustment Cancelled	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x04D3	Credit Adjustment Rejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x04D4	Change Payment Mode Received	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x04D5	Change Payment Mode Actioned	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x04D6	Change Payment Mode Cancelled	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x04D7	Change Payment Mode Rejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x04D8	GetPrepaySnapshotReceived	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x04D9	GetPrepaySnapshotActioned	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x04DA	GetPrepaySnapshotCancelled	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x04DB	GetPrepaySnapshotRejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x04DC	GetTopupLogReceived	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x04DD	GetTopupLogActioned	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x04DE	GetTopupLogCancelled	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x04DF	GetTopupLogRejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x04E0	Set Low Credit Warning Level Received	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x04E1	Set Low Credit Warning Level Actioned	8-bit Bitmap	0x00 – 0xFF	Read only		O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x04E2	Set Low Credit Warning Level Cancelled	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x04E3	Set Low Credit Warning Level Rejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x04E4	GetDebtRepayLog Received	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x04E5	GetDebtRepayLog Actioned	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x04E6	GetDebtRepayLog Cancelled	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x04E7	GetDebtRepayLog Rejected	8-bit Bitmap	0x00 – 0xFF	Read– only		O
0x04E8	SetMaximumCreditLimit Received	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x04E9	SetMaximumCreditLimit Actioned	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x04EA	SetMaximumCreditLimit Cancelled	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x04EB	SetMaximumCreditLimit Rejected	8-bit Bitmap	0x00 – 0xFF	Read– only		O
0x04EC	SetOverallDebtCap Received	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x04ED	SetOverallDebtCap Actioned	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x04EE	SetOverallDebtCap Cancelled	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x04EF	SetOverallDebtCap Rejected	8-bit Bitmap	0x00 – 0xFF	Read– only		O
0x04F0 - 0x04FE	RESERVED					
0x04FF	Reserved for Prepayment cluster Group ID	--	--	Read only	-	O

12541

12542 D.10.3.2.5.1 Event Configuration Attributes

12543 The attributes in this set allow a server device to configure how an event is handled when
12544 triggered. All attributes in this set are bitmaps as defined in Table D-177.

12545 D.10.3.2.6 **Calendar Event Configuration Attribute Set**

12546 The following attributes allow events related to calendars to be configured.

12547

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

12548 **Table D-181– Calendar Event Configuration Attribute Set**

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0500	Calendar Cluster Not Found	8-bit Bitmap	0x00 – 0xFF	Read–only		O
0x0501	Calendar Change Passive Activated	8-bit Bitmap	0x00 – 0xFF	Read–only		O
0x0502	Calendar Change Passive Updated	8-bit Bitmap	0x00 – 0xFF	Read–only		O
0x0503 - 0x05BF	RESERVED					
0x05C0	PublishCalendar Received	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x05C1	PublishCalendar Actioned	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x05C2	PublishCalendar Cancelled	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x05C3	PublishCalendar Rejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x05C4	Publish Day Profile Received	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x05C5	Publish Day Profile Actioned	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x05C6	Publish Day Profile Cancelled	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x05C7	Publish Day Profile Rejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x05C8	Publish Week Profile Received	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x05C9	Publish Week Profile Actioned	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x05CA	Publish Week Profile Cancelled	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x05CB	Publish Week Profile Rejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x05CC	Publish Seasons Received	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x05CD	Publish Seasons Actioned	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x05CE	Publish Seasons Cancelled	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x05CF	Publish Seasons Rejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x05D0	Publish Special Days Received	8-bit Bitmap	0x00 – 0xFF	Read only		O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x05D1	Publish Special Days Actioned	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x05D2	Publish Special Days Cancelled	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x05D3	Publish Special Days Rejected	8-bit Bitmap	0x00 – 0xFF	Read– only		O
0x05D6 - 0x05FE	RESERVED					
0x05FF	Reserved for Calendar cluster Group ID	--	--	Read only	-	O

12549

12550 D.10.3.2.6.1 Event Configuration Attributes

12551 The attributes in this set allow a server device to configure how an event is handled when
12552 triggered. All attributes in this set are bitmaps as defined in Table D-177.

12553 D.10.3.2.7 **Device Management Event Configuration Attribute Set**

12554 The following attributes allow events related to device management to be configured.

12555 **Table D-182– Device Management Event Configuration Attribute Set**

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0600	Password1Change	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0601	Password2Change	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0602	Password3Change	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0603	Password4Change	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0604	EventLogCleared	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0605 – 0x060F	RESERVED					
0x0610	ZigBee APS Timeout	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0611	ZigBee IEEE Transmission Failure Over Threshold	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0612	ZigBee IEEE Frame Check Sequence Threshold	8-bit Bitmap	0x00 – 0xFF	Read only		O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0613	Error Certificate	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0614	Error Signature	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0615	Error Program Storage	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0616 – 0x06BF	RESERVED					
0x06C0	Publish CoT Received	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x06C1	Publish CoT Actioned	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x06C2	Publish CoT Cancelled	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x06C3	Publish CoT Rejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x06C4	Publish CoS Received	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x06C5	Publish CoS Actioned	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x06C6	Publish CoS Cancelled	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x06C7	Publish CoS Rejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x06C8	Change Password Received	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x06C9	Change password Actioned	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x06CA	Change Password Cancelled	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x06CB	Change Password Rejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x06CC	SetEventConfiguration Received	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x06CD	SetEventConfiguration Actioned	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x06CE	SetEventConfiguration Cancelled	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x06CF	SetEventConfiguration Rejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x06D0	UpdateSiteID Received	8-bit Bitmap	0x00 – 0xFF	Read only		O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x06D1	UpdateSiteID Actioned	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x06D2	UpdateSiteID Cancelled	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x06D3	UpdateSiteID Rejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x06D4	UpdateCIN Received	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x06D5	UpdateCIN Actioned	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x06D6	UpdateCIN Cancelled	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x06D7	UpdateCIN Rejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x06D8- 0x06FE	RESERVED					
0x06FF	Reserved for Device Management cluster Group ID	--	--	Read only	-	O

12556

12557 D.10.3.2.7.1 Event Configuration Attributes

12558 The attributes in this set allow a server device to configure how an event is handled when
12559 triggered. All attributes in this set are bitmaps as defined in Table D-177.

12560 D.10.3.2.8 **Tunnel Event Configuration Attribute Set**

12561 The following attributes allow events related to tunneling to be configured.

12562 **Table D-183– Tunneling Event Configuration Attribute Set**

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0700	Tunneling Cluster Not Found	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0701	Unsupported Protocol	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0702	IncorrectProtocol	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0703- 0x07BF	RESERVED					

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x07C0	RequestTunnel Command Received	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x07C1	RequestTunnel Command Rejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x07C2	RequestTunnel Command Generated	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x07C3	CloseTunnel Command Received	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x07C4	CloseTunnel Command Rejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x07C5	CloseTunnel Command Generated	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x07C6	TransferData Command Received	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x07C7	TransferData Command Rejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x07C8	TransferData Command Generated	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x07C9	TransferDataError Command Received	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x07CA	TransferDataError Command Rejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x07CB	TransferDataError Command Generated	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x07CC	AckTransferData Command Received	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x07CD	AckTransferData Command Rejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x07CE	AckTransferData Command Generated	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x07CF	ReadyData Command Received	8-bit Bitmap	0x00 – 0xFF	Read only		O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x07D0	ReadyData Command Rejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x07D1	ReadyData Command Generated	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x07D2	GetSupportedTunnelProtocols Command Received	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x07D3	GetSupportedTunnelProtocols Command Rejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x07D4	GetSupportedTunnelProtocols Command Generated	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x07D5-0x07FE	RESERVED					
0x07FF	Reserved for Tunnel cluster Group ID	--	--	Read only	-	O

12563

12564 D.10.3.2.8.1 Event Configuration Attributes

12565 The attributes in this set allow a server device to configure how an event is handled when
12566 triggered. All attributes in this set are bitmaps as defined in Table D-177.

12567 D.10.3.2.9 **OTA Event Configuration Attribute Set**

12568 The following attributes allow events related to OTA to be configured.

12569 **Table D-184– OTA Event Configuration Attribute Set**

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0800	FirmwareReadyForActivation	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0801	FirmwareActivated	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0802	Firmware Activation Failure	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0803	Patch Ready For Activation	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0804	Patch Activated	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0805	Patch Failure	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0806 - 0x08BF	RESERVED					

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x08C0	Image Notify Command Received	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x08C1	Image Notify Command Rejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x08C2	Query Next Image Request Generated	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x08C3	Query Next Image Response Received	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x08C4	Query Next Image Response Rejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x08C5	Image Block Request Generated	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x08C6	Image Page Request Generated	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x08C7	Image Block Response Received	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x08C8	Image Block Response Rejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x08C9	Upgrade End Request Generated	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x08CA	Upgrade End Response Received	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x08CB	Upgrade End Response Rejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x08CC	Query Specific File Request Generated	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x08CD	Query Specific File Response Received	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x08CE	Query Specific File Response Rejected	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x08CF – 0x08FE	RESERVED					
0x08FF	Reserved for OTA cluster Group ID	--	--	Read only	-	O

12570

12571 D.10.3.2.9.1 Event Configuration Attributes

12572 The attributes in this set allow a server device to configure how an event is handled when
12573 triggered. All attributes in this set are bitmaps as defined in Table D-177.
12574

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

12575 #### D.10.3.3 Commands Received

12576 The client receives the cluster-specific response commands detailed in sub-clause D.10.2.4.

12577 #### D.10.3.4 Commands Generated

12578 The client generates the cluster-specific commands detailed in sub-clause D.10.2.3, as required by
12579 the application.
12580

12581 ## D.10.4 Application Guidelines

12582 #### D.10.4.1 Passwords

12583 The use of Password within this cluster could also be viewed as PIN codes. The current use case
12584 for Passwords is to cover either the consumer PIN code, or to secure the engineer maintenance
12585 screens found on a metering device.

12586 #### D.10.4.2 Consumer Password Use Case

12587 The Password or (normally) PIN code is part of the application and, as such, not a data item that
12588 would need to be supplied by the HES or held by the server. There is normally a screen on a
12589 device to be able to set or enter a new password. The main use case for the consumer is therefore
12590 to instruct the device to reset the PIN so that the consumer can again gain access to the IHD
12591 screens. The server is therefore only required to hold an access list that contains the device
12592 EUI64 address.

12593 #### D.10.4.3 Engineer Password Use Case

12594 The Engineer password is normally used to access maintenance screens on meters, so that key
12595 functions can be secured and only accessed by an authorised personal, or for data that is sensitive
12596 to the operation of the device such as Joining or Leaving the HAN.

12597 #### D.10.4.4 Password Security Recommendations

12598 If additional security is required by the application, it is recommended that the password octet-
12599 string is sent as a hashed value, using MMO hashing to create a 48-bit hashed value. The Device
12600 Management Cluster is APS Secured and the password is always unicast to the individual device.
12601 An Access Control List within the server is recommended, to allow for the management of the
12602 passwords against the EUI-64 address of the device. However, the actual format of the password
12603 octet-string is down to the implementation requirements of the system.

12604 If a device is unable to use a password then NO data should be shown that has been deemed to be
12605 password protected.

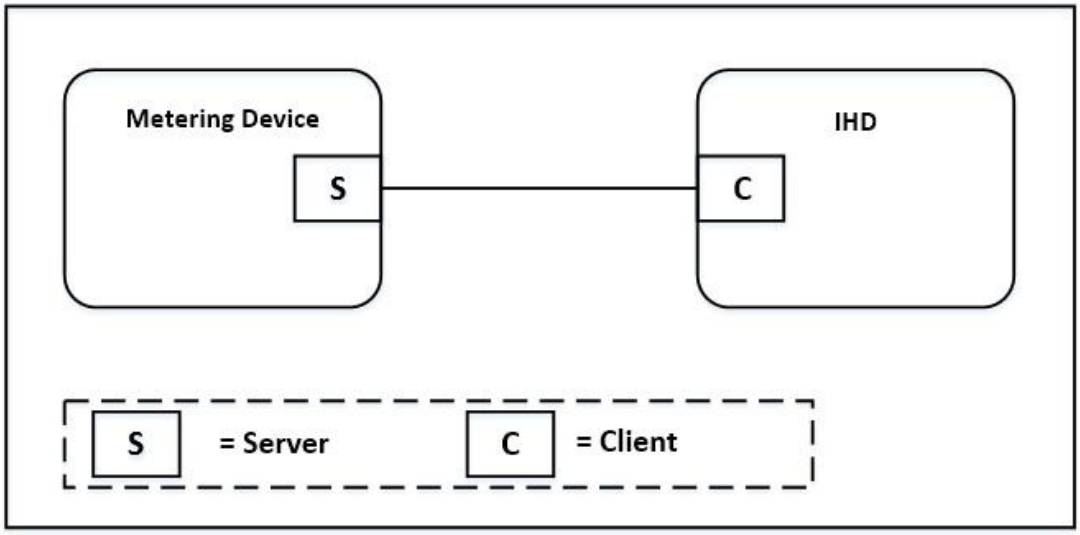
12606

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

12607 ## D.11 Events Cluster

12608 ### D.11.1 Overview

12609 This cluster provides an interface for passing event information between ZigBee devices. Events
12610 are generated and logged by a server device and read by a client device.



12611

12612 **Figure D-169– Event Cluster Client Server Example**

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

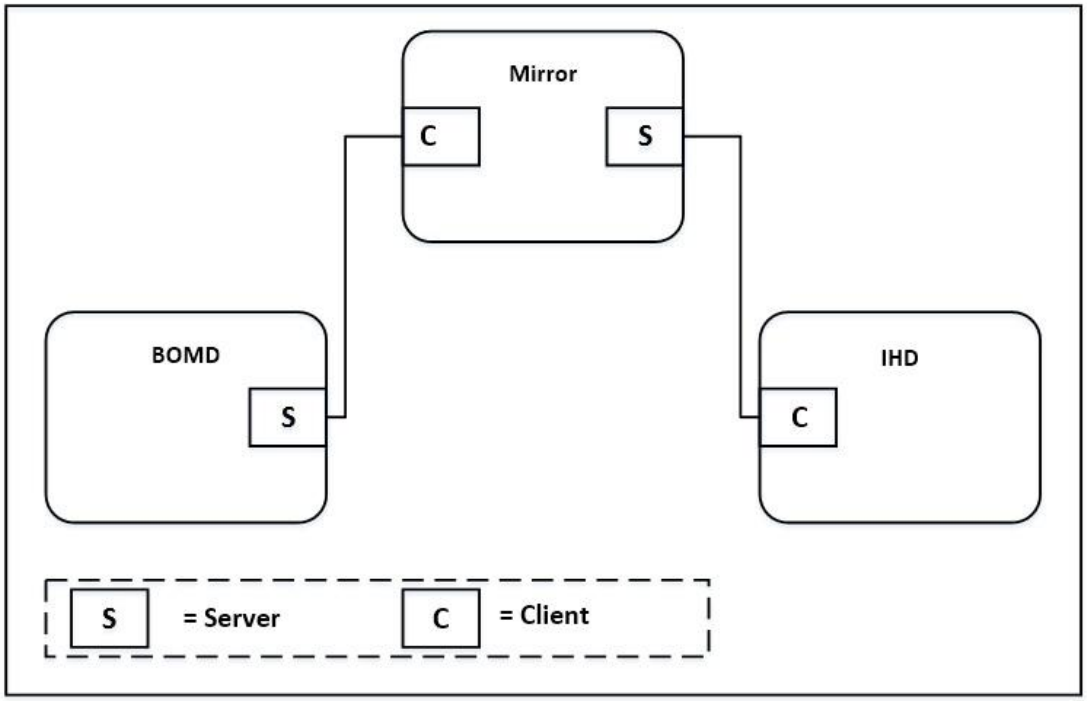


Figure D-170– Mirrored BOMD Event Cluster Client Server Example

D.11.2 Server

D.11.2.1 Dependencies

- Events carried using this cluster include a timestamp with the assumption that target devices maintain a real time clock. Devices can acquire and synchronize their internal clocks via the ZCL Time server.

- A server device supporting this cluster should also support the Device Management Cluster in order to allow events to be configured over the air.

- In order that Events Cluster client devices are able the receive events published from an Events Cluster server on a BOMD, the BOMD mirror should support both an Events cluster client and server. The BOMD should publish events to the mirror and the mirror should, if required (based on the control flags in the *PublishEvent* command), publish events to all bound Events Cluster client devices.

- Events Cluster client devices wishing to receive events published from a BOMD shall bind to the Events Cluster server on the BOMD mirror.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

- A Mirror may store the events pushed from a BOMD, effectively mirroring the BOMD event logs. The Mirror may also support the reading and clearing of event logs by Events Cluster client devices.

- How events are internally stored within an Events Cluster server device is out of scope of this specification.

D.11.2.2 Attributes

None

D.11.2.3 Commands Received

Table D-185 lists the cluster specific commands that are received by the server.

Table D-185– Cluster Specific Commands Received by the Server

Command Identifier Field Value	Description	Mandatory / Optional
0x00	GetEventLog	O
0x01	Clear Event Log Request	O

D.11.2.3.1 **Get Event Log Command**

The *GetEventLog* command allows a client to request events from a server's event logs. One or more *PublishEventLog* commands are returned on receipt of this command.

The *LogID* sub-field, in conjunction with the *Event ID* field, shall provide the filtering to enable the desired event(s) to be identified. The following examples show the usage of these 2 fields:-

1 Get all events from the Security Event Log (Log ID = Security (4), Event ID = 0x0000)

2 Get all events from all logs (Log ID = 0, Event ID = 0x0000)

3 Get all occurrences of a specific event 0x1111 from all logs (Log ID = 0, Event ID = 0x1111)

4 Get all occurrences of a specific event 0x1111 from the Security Event log (Log ID = Security (4), Event ID = 0x1111).

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

12654 D.11.2.3.1.1 Payload Format

Octets	1	2	4	4	1	2
Data Type	8-bit Bitmap	Unsigned 16-bit Integer	UTC Time	UTC Time	Unsigned 8-bit Integer	Unsigned 16-bit Integer
Field Name	Event Control/ Log ID (M)	Event ID (M)	Start Time (M)	End Time (M)	Number of Events (M)	Event Offset (M)

12655 **Figure D-171– Get Event Log Command Payload**

12656 D.11.2.3.1.2 Payload Details

12657 **Event Control/Log ID (mandatory):** The least significant nibble is an enumeration indicating
12658 the Log ID (see Table D-186). The most significant nibble is a bitmap indicating control options
12659 (see Table D-187):

12660 **Table D-186– Log ID Enumeration**

Bit	Description	
0-3	Enumerated Value	Description
	0x0	All logs
	0x1	Tamper Log
	0x2	Fault Log
	0x3	General Event Log
	0x4	Security Event Log
	0x5	Network Event Log
	0x6-0xF	Reserved

12661

12662 **Table D-187– Event Control Bitmap**

Bit	Description
4	0- retrieve the minimal information per event (Event ID and Time) 1-retrieve the full information per event (Event ID, Time and Octet string, if available)
5-7	Reserved

12663

12664 **Event ID (mandatory):** The *Event ID* specifies a particular event to be queried; a value of
12665 0x0000 is reserved to indicate all Event IDs. The Event IDs for the Smart Energy profile are
12666 detailed in tables Table D-176 to Table D-184.

12667 *Note: If event configuration is supported via the device management cluster the ZigBee Event*
12668 *IDs are defined in tables* Table D-176 *to* Table D-184.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

12669 **Start Time (mandatory):** This field specifies the start time (earliest time) of the range of events
12670 to be returned. Events that match the search criteria and have a timestamp **greater than or equal**
12671 **to** the start time shall be returned.

12672 **End Time (mandatory):** specifies the end time (latest time) of the range of events to be reported
12673 in the response. Events that match the search criteria and have a timestamp **less than** the
12674 specified end time shall be returned. Events with a timestamp **equal to** that of the *End Time* shall
12675 not be returned; this ensures that, in the case where the *End Time* is set to the current time, events
12676 generated whilst reading the event log are not included in the response.

12677 **Number of Events (mandatory):** This parameter indicates the maximum number of events
12678 requested i.e. the maximum number of events that the client is willing to receive; the value 0x00
12679 indicates all events that fall into the defined criteria.

12680 **Event Offset (mandatory):** The *Event Offset* field provides a mechanism to allow client devices
12681 to page through multiple events which match a given search criteria. As an example, a client
12682 device requests two events from a given search criteria with an *Event Offset* of 0. The server
12683 returns the two most recent events (events 1 and 2) in a *PublishEvent* command and indicates
12684 that 4 events match the given criteria. The client re-sending the original *Get Event Log* command,
12685 but with the *Event Offset* field now set to 2, shall result in the server returning events 3 and 4.

12686 D.11.2.3.1.3 Effect on Receipt

12687 On receipt of this command, the device shall respond with a *PublishEventLog* command A ZCL
12688 Default Response with status NOT_FOUND shall be returned if no events match the given
12689 search criteria.

12690

12691 D.11.2.3.2 **Clear Event Log Request Command**

12692 This command requests that an Events server device clear the specified event log(s). The Events
12693 server device SHOULD clear the requested events logs, however it is understood that market
12694 specific restrictions may be applied to prevent this.

12695 D.11.2.3.2.1 Payload Format

Octets	**1**
Data Type	8-bit BitMap
Field Name	Log ID (M)

12696 **Figure D-172– Clear Event Log Request Command Payload**

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

12697

12698 D.11.2.3.2.2 Payload Details

12699 **Log ID (mandatory):** The least significant nibble specifies the Log to be cleared (see Table
12700 D-186). The most significant nibble is reserved.

12701 D.11.2.3.2.3 Effect on Receipt

12702 On receipt of this command, a device supporting the Events cluster as a server should clear the
12703 specified event logs. A *Clear Event Log Response* command shall be generated, indicating which
12704 event logs have been successfully cleared.

12705

12706 ### D.11.2.4 Commands Generated

12707 Table D-188 lists the cluster specific commands that are generated by the server.

12708 **Table D-188– Cluster Specific Commands Generated by the Server**

Command Identifier Field Value	Description	Mandatory / Optional
0x00	Publish Event	O
0x01	Publish EventLog	O
0x02	Clear Event Log Response	O

12709

12710 D.11.2.4.1 **Publish Event Command**

12711 This command is generated upon an event trigger from within the reporting device and if enabled
12712 by the associated Event Configuration (bitmap) attribute in the Device Management cluster (see
12713 Table D-177 for further information).

12714 D.11.2.4.1.1 Payload Format

Octets	**1**	**2**	**4**	**1**	**1..255**
Data Type	8-bit Bitmap	Unsigned 16-bit Integer	UTC Time	8-bit Bitmap	Octet String
Field Name	Log ID (M)	Event ID (M)	Event Time (M)	Event Control (M)	Event Data (M)

12715 **Figure D-173– Publish Event Command Payload**

12716 D.11.2.4.1.2 Payload Details

12717 **Log ID (mandatory):** The least significant nibble is an enumeration indicating the Log ID (see
12718 Table D-186). The most significant nibble is reserved.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

12719 **Event ID (mandatory):** The *Event ID* specifies a particular event. If event configuration is
12720 supported (via the Device Management cluster), the ZigBee Event IDs are as defined in Table
12721 D-176 to Table D-184.

12722 **Event Time (mandatory)**: The timestamp of the event occurrence in UTC format.

12723 **Event Control (mandatory)**: An 8-bit bitmap specifying actions to be taken regarding this
12724 event:

12725 **Table D-189– Event Action Control Bitmap**

Bit	Description (if set)
0	Report Event to HAN devices – this flag indicates that the event is intended for the HAN; the event should be published to all bound Events cluster client devices. If the event is generated by a BOMD and received by a mirror, the mirror should publish this event to all bound Events cluster clients.
1	Report Event to the WAN – this flag indicates that the event is intended for the WAN; if the receiving device is capable, it should report this event to the WAN.
2-7	Reserved

12726

12727 **Event Data (mandatory):** A variable length octet string array used to hold additional
12728 information captured when the event occurred. The first element (element 0) of the array
12729 indicates the length of the string, NOT including the first element.

12730

12731 D.11.2.4.2 **Publish Event Log Command**

12732 This command is generated on receipt of a *Get Event Log* command. The command shall return
12733 the most recent event first, up to the number of events requested.

12734 D.11.2.4.2.1 Payload Format

Octets	2	1	1	1..xx
Data Type	Unsigned 16-bit Integer	Unsigned 8-bit Integer	Unsigned 8-bit Integer	
Field Name	Total Number of Matching Events (M)	Command Index (M)	Total Commands (M)	Log Payload (M)

12735 **Figure D-174– Publish Event Log Command Payload**

12736 D.11.2.4.2.2 Payload Details

12737 **Total Number of Matching Events (mandatory):** This field indicates the total number of
12738 events found which match the search criteria received in the associated *Get Event Log* command.
12739 The value of this field may be greater than the total number of events requested; if this is the case

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

12740 then further events may be retrieved using the *Event Offset* field of the *Get Event Log* command
12741 (see D.11.2.3.1).

12742 **Command Index (mandatory):** In the case where the entire number of events being returned
12743 does not fit into a single message, the *Command Index* is used to count the required number of
12744 *Publish Event Log* commands. The *Command Index* starts at 0 and is incremented for each
12745 command returned due to the same *Get Event Log* command.

12746 **Total Commands (mandatory):** This parameter indicates the total number of *Publish Event Log*
12747 commands that are required to return the requested event logs.

12748 ***D.11.2.4.2.2.1 Log Payload***

12749 The *Log Payload* is a series of events and associated data. The event payload consists of the
12750 logged events as detailed in Figure D-175:

Octets	**1**	**1**	**2**	**4**	**1..255**	**…**	**1**	**2**	**4**	**1..255**
Data Type	8-bit Bitmap	8-bit Bitmap	Unsigned 16-bit Integer	UTC Time	Octet String	…	8-bit Bitmap	Unsigned 16-bit Integer	UTC Time	Octet String
Field Name	Number of Events / Log Payload Control(M)	Log ID (M)	Event ID (M)	Event Time (M)	Event Data (M)	…	Log ID (O)	Event ID n (O)	Event Time n (O)	Event Data n (O)

12751 **Figure D-175– Publish Event Log Sub-Payload**

12752 **Number of Events /Log Payload Control (mandatory):** This field is split into two parts; the
12753 least significant nibble represents the *Log Payload Control* as defined in Table D-190, whilst the
12754 most significant nibble indicates the number of events contained within the log payload of this
12755 command. Note that an event which crosses a payload boundary is considered to be 1 event in
12756 the log payload. Wherever possible events SHOULD NOT be sent across payload boundaries.

12757 **Table D-190– Log Payload Control Bitmap**

Bit	**Description**
0	0 - Events do not cross frame boundary 1 – An event in this log payload does cross a payload frame boundary
1 – 3	Reserved

12758

12759 **Log ID (mandatory):** The least significant nibble is an enumeration indicating the Log ID (see
12760 Table D-186). The most significant nibble is reserved.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Event ID: The *Event ID* specifies a particular event. If event configuration is supported (via the Device Management cluster), ZigBee-specified Event IDs are as defined in Table D-176 to Table D-184.

Event Time: The timestamp of the event occurrence in UTC format.

Event Data: A variable length octet string array used to hold additional information captured when the event occurred. The first element (element 0) of the array indicates the length of the string, NOT including the first element. This field should contain a single octet of 0x00 when 'minimal information' is requested in the associated *Get Event Log* command (see D.11.2.3.1.2 for further details).

D.11.2.4.3 **Clear Event Log Response Command**

This command is generated on receipt of a *Clear Event Log Request* command.

D.11.2.4.3.1 Payload Format

Octets	**1**
Data Type	8 Bit Bitmask
Field Name	ClearedEventsLogs (M)

Figure D-176– Clear Event Log Response Command Payload

D.11.2.4.3.2 Payload Details

ClearedEventsLogs (mandatory): This 8-bit BitMask indicates which logs have been cleared, as detailed in Table D-191.

Note: It is understood that certain markets may require that event logs cannot be cleared; this BitMask provides a method for the server device to indicate which logs have been successfully cleared.

Table D-191– ClearedEventsLogs Bitmap

Bit	**Description**
0	0 – All Logs NOT Cleared 1 - All Logs Cleared
1	0 - Tamper Log NOT Cleared 1 - Tamper Log Cleared
2	0 - Fault Log NOT Cleared 1 - Fault Log Cleared
3	0 - General Event Log NOT Cleared

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

	1 - General Event log Cleared
4	0 - Security Event Log NOT Cleared 1 - Security Log Cleared
5	0 - Network Event Log NOT cleared 1 - Network Event Log cleared
6-7	Reserved

12782

12783

12784 ## D.11.3 Client

12785 ### D.11.3.1 Dependencies

12786 Events carried using this cluster include a timestamp with the assumption that target devices
12787 maintain a real time clock. Devices can acquire and synchronize their internal clocks via the ZCL
12788 Time server.

12789 ### D.11.3.2 Attributes

12790 None.

12791 ### D.11.3.3 Commands Received

12792 See section D.11.2.4.

12793 ### D.11.3.4 Commands Generated

12794 See section D.11.2.3.

12795

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

D.12 Energy Management Cluster

Note: *The Energy Management Cluster description in this revision of this specification is provisionary and not certifiable. This feature set may change before reaching certifiable status in a future revision of this specification.*

D.12.1 Overview

This cluster provides a way of modifying DRLC events, energy consumption behaviour and querying the status of DRLC events.

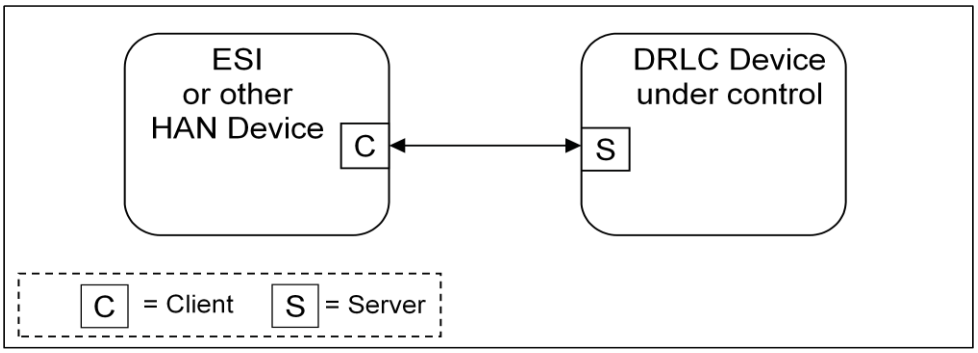


Figure D-177– Energy Management Cluster

Note that the ESI is defined as the Client. The DRLC device is a Server in this case; it holds the attributes and receives commands.

D.12.2 Server

D.12.2.1 Dependencies

A server device shall support the DRLC cluster as a client.

D.12.2.2 Attributes

Table D-192– Energy Management Server Cluster Attributes

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0000	LoadControlState	8-bit BitMap	0x00 to 0xFF	Read Only	0x00	M
0x0001	*CurrentEventID*	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFE	Read Only	0xFFFFFFFF	M

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

0x0002	*CurrentEventStatus*	8-bit BitMap	0x00 to 0xFF	Read Only	0x00	M
0x0003	*ConformanceLevel*	Unsigned 8-bit Integer	0x00 to 0x07	Read/ Write or Read Only	0x00	M
0x0004	*MinimumOffTime*	Unsigned 16-bit Integer	0x0000 to 0x0258 (10 minutes) or 0xFFFF	Read/ Write or Read Only	TBD	M
0x0005	*MinimumOnTime*	Unsigned 16-bit Integer	0x0000 to 0x0258 (10 minutes) or 0xFFFF	Read/ Write or Read Only	TBD	M
0x0006	*MinimumCyclePeriod*	Unsigned 16-bit Integer	0x0000 to 0x4650 (5 hours) or 0xFFFF	Read/ Write or Read Only	TBD	M

12814

12815 D.12.2.2.1 **Load Control State Attribute**

12816 This attribute shall be a BitMap showing the current state of the device. The attribute shall be
12817 read-only.

12818 **Table D-193– Load Control State BitMap/Encoding**

Bit	Description
0	Relay Open or Consumption Interupted
1	Event In Progress
2	Power Stabilizing
3	Other Load Reduction
4	Current Flow or Consuming Commodity
5	Load Call
6 – 7	Reserved

12819

12820 **Relay Open or Consumption Interrupted:** The device being controlled has been turned off and
12821 prevented from consuming electric power or another commodity, either by opening a relay or
12822 some other means.

12823 **Event In Progress:** There is an event in progress. The current time is between the Effective Start
12824 Time and Effective End Time of the event. When this bit is set, the *CurrentEventStatus* attribute
12825 is valid.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

12826 **Power Stabilizing:** The device has automatically reduced consumption of the commodity for an
12827 automatic reason, to aid in the stability of the system. Possible actions and reasons might be:

12828 • A random wait after a power outage before starting an electric motor.

12829 • Duty cycling heating after a long power outage, to prevent all electric heaters being on at the
12830 same time.

12831 • Turning off a furnace if the gas pressure drops, open electric relay if the frequency or voltage
12832 are off, etc.

12833 **Other Load Reduction:** This bit indicates the device has automatically reduced consumption of
12834 the commodity for some non-consumer-initiated reason. Possible automatic actions and reasons
12835 might be:

12836 • A PCT raising the cooling set point in response to an increase in price.

12837 • A Load Control Device turning off a pool pump or other non-essential appliance when
12838 energy credits are low.

12839 **Current Flow or Consuming Commodity:** This bit indicates that the device is currently
12840 consuming the commodity. The bit not being set means either no commodity being consumption
12841 or the device does not have the ability to detect consumption. Support is optional. The bit shall be
12842 cleared if not supported.

12843 **Load Call:** This bit is set if there is currently no consumption but the device under control would
12844 consume power if able to. Support is optional. The bit shall be cleared if not supported.
12845

12846 D.12.2.2.2 **CurrentEventID Attribute**

12847 If an event is in progress (current time is between the Effective Start Time and Effective End
12848 Time of an event), this attribute SHALL indicate the *Issuer Event ID* of the active event. The
12849 invalid Event ID 0xFFFFFFFF SHALL be used when an event is NOT active. The attribute
12850 SHALL be read-only.

12851 D.12.2.2.3 **CurrentEventStatus Attribute**

12852 This attribute represents the value returned in the *Event Control* field of the latest *Report Event*
12853 *Status* command (see D.2.3.3.1). This attribute provides a mechanism to allow a remote device to
12854 query whether this client device is currently participating in a load control event. Typical use of
12855 this attribute is expected to be for the refresh of UIs. The attribute SHALL be read-only.

12856 In conjunction with this new attribute, the BitMap defined for the *Event Control* field of both
12857 *Load Control Event* and *Report Event Status* commands SHALL be extended to cater for the

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

12858 following additional control options (note that only *Report Event Status* commands will use these
12859 additional bits):

12860

12861 **Table D-194– Current Event Status BitMap/Encoding**

Bit	Description
0	Randomized Start Time
1	Randomized Duration
2	Extended Bits Present
3	Event Active
4	Device participating in Event (not opted out)
5	Reducing Load
6	On at end of Event
7	Reserved

12862

12863 **Randomized Start Time Attribute:** Set if the current event had a randomized start time.

12864 **Randomized Duration:** Set if the duration of the current event is randomized.

12865 **Extended Bits Present:** This bit will always return 1. This allows the field to be used in DRLC
12866 *Event Status* messages.

12867 **Event Active:** Set if the current time lies between the Effective Start Time and Effective End
12868 Time of the event identified by the *CurrentEventID* attribute (Note that, if this bit is not set when
12869 the *CurrentEventStatus* attribute is read, none of the other bits are valid since there is no current
12870 event).

12871 **Device Participating in Event:** Set if the device is (or will be when the event starts)
12872 participating in the event (i.e. not opted out) Note that a device can participate in an event and
12873 not actually do anything, i.e. this bit could be set while the *Load Reduction* bit is not.

12874 **Reducing Load:** Set if the device is currently shedding load in response to a DR event. Set if an
12875 active load control event is duty cycling and currently off, or if the new set points or offsets are
12876 lowering demand. Clear if the device is participating in an event, but is currently in the On
12877 portion of the duty cycle. Clear if none of the fields of the DRLC event applied to the device (e.g.
12878 only setpoints were set but the device is a pool pump). Support is optional. The bit shall be set if
12879 not supported.

12880 **On at End of Event:** Set if the device will return to using a normal load after the event has
12881 completed. For example, this would be False if the device supported the On/Off cluster ([B1])

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

and the *OnOff* attribute was set to OFF. It would also be False if the device was a PCT, the current temperature was lower than the set point and the device was in cooling mode. Note that, if an event is not active, this value may not be reliable. Support is optional; if unsupported, the bit shall be set.

D.12.2.2.4 **Conformance Level Attribute**

This is the minimum criticality level of a DRLC event that the device will observe. Events with a criticality level lower than the *Conformance Level* will be opted out. If a change in the *Conformance Level* results in a currently running event being opted out (or in), a *ReportEventStatus* message must be generated. If the event(s) has(have) not started, the device may send a *ReportEventStatus* immediately or when the event starts. The attribute shall be read/write if supported, with a valid range of 1 -7, or 0 and read-only if unsupported.

D.12.2.2.5 **MinimumOffTime, MinimumONTime & MinimumCyclePeriod Attributes**

These attributes are measured in seconds and used to determine the duty cycling times. The *MinimumOffTime* and *MinimumOnTime* may also be enforced when starting or ending a DRLC event, or when the On/Off cluster opens or closes the relay. From these values, the DutyOnTime and DutyOffTime times are calculated.

MinimumCyclePeriod is the shortest cycling period allowed for duty cycling, while *MinimumOnTime* and *MinimumOffTime* are the shortest times the device should be allowed on or off. The on/off minimums are to protect devices that can be damaged by being on or off for short time periods, while the *MinimumCyclePeriod* prevents the relay from cycling too quickly. The attribute values shall be 0xFFFF and read-only if unsupported, or read/write if supported. A *MinimumCyclePeriod* of 0 indicates that duty cycling shall be disabled; an event calling for any duty cycling will then just turn the device off for the duration of the event.

Algorithm for calculating DutyOnTime and DutyOffTime

```
DutyOnTime: the amount of time in a duty cycle period that the device is on
DutyOffTime:the amount of time in a duty cycle period that the device is off

DutyOnTime = MinimumCyclePeriod * dutyCycleApplied / 100
DutyOffTime = MinimumCyclePeriod - DutyOnTime
//NOTE except for dutyCycleApplied near 99% or 1% we are likely done

If (DutyOnTime < MinimumONTime)
    DutyOnTime = MinimumONTime
    DutyOffTime = MinimumONTime * (100 – dutyCycleApplied)/ dutyCycleApplied

If (DutyOffTime < MinimumOffTime)
    DutyOffTime = MinimumOffTime
    DutyONTime = MinimumOffTime * dutyCycleApplied/ (100 – dutyCycleApplied)
```

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

12922

12923

12924 ### D.12.2.3 Attribute Reporting

12925 Attribute reporting is not expected to be used for this cluster.

12926 ### D.12.2.4 Commands Received

12927 The command IDs received by the Energy Management cluster server are listed in Table D-195.

12928 **Table D-195– Received Command IDs for the Energy Management Server**

Command Identifier Field Value	Description	Mandatory/ Optional
0x00	*ManageEvent*	M

12929

12930 D.12.2.4.1 **Manage Event Command**

12931 The *Manage Event* command allows a remote device (such as an IHD or web portal) to change
12932 the behavior of a DRLC cluster client when responding to a DRLC *Load Control Event*.

12933 D.12.2.4.1.1 Payload Format

Octets	4	2	1	1
Data Type	Unsigned 32-bit Integer	16-bit BitMap	Unsigned 8-bit Integer	Unsigned 8-bit Integer
Field Name	Issuer Event ID (M)	Device Class (M)	Utility Enrollment Group (M)	Action(s) Required

12934 **Figure D-178– *Manage Event* Command Payload**

12935 D.12.2.4.1.2 Payload Details

12936 **Issuer Event ID (mandatory):** Unique identifier generated by the Energy provider. The value of
12937 this field allows the command to be matched with a specific Load Control Event. A value of
12938 0xFFFFFFFF indicates the current running event.

12939 **Device Class (mandatory):** Bit encoded field representing the DRLC client Device Class to
12940 apply the current *Load Control Event*. Each bit, if set individually or in combination, indicates
12941 the class of device(s) the *Manage Event* command is directed at. If the requested device class(es)
12942 is not implemented by the receiving device, the command should be ignored.

12943 **Utility Enrollment Group (UEG) (mandatory):** The DRLC client *Utility Enrolment Group*
12944 field can be used in conjunction with the *Device Class* bits. It provides a mechanism to direct the

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Manage Event command to groups of devices. If the requested UEG is non-zero and does not match the UEG of the receiving device, the command should be ignored.

Action(s) Required: Bit encoded field indicating the action(s) to be carried out in regard to the associated event. Bits, set individually or in combination, shall be provided for the following actions:

Table D-196– Action(s) Required BitMap/Encoding

Bit	Description
0	Opt Out of Event
1	Opt Into Event
2	Disable Duty Cycling
3	Enable Duty Cycling
4 -7	Reserved

If the *Manage Event* command is valid, a *Report Event Status* command will be returned regardless of whether any bits are set.

Notes:

1 If the load control event is mandatory and the device is Smart Energy compliant, requests to opt-out shall be ignored. Opt-in shall always be honored.

2 If the device does not support duty cycling, or the load control event did not request duty cycling, requests to modify duty cycling should be ignored, however a *Report Event Status* command should still be sent.

3 If the *ManageEvent* command action was to opt-out or opt-in, the *Event Status* field in the associated *Report Event Status* response shall be 0x04 (User Opt-out) or 0x05 (User Opt-in). For all other actions, the *Event Status* field in the associated *Report Event Status* response shall be set to 0x01 (if the load control event has not yet started), 0x02 (if the event is running and opted in) or 0x04 (if the event is running but currently opted out; this is to allow for backwards compatibility).

The rationale behind disabling duty cycling is to give the consumer more control over how much energy to shed during the event, without necessarily requiring them to place their device in a state where it will be off after the event. The response to disabling duty cycling will be seen as a value of 0 in the duty cycle applied field of the *Report Event Status* message.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

12971 ### D.12.2.5 Commands Generated

12972 The command IDs generated by the Energy Management cluster server are listed in Table
12973 D-197.

12974 **Table D-197– Generated Command IDs for the Energy Management Server**

Command Identifier Field Value	Description	Mandatory/ Optional
0x00	*ReportEventStatus*	M

12975

12976 #### D.12.2.5.1 **Report Event Status Command**

12977 This command is reused from the DRLC cluster. This command is generated in response to the
12978 *Manage Event* command. It is likely that a *Manage Event* command will also change a DRLC
12979 event causing the DRLC client to send a DRLC *Report Event Status* message as well (see
12980 D.2.3.3.1).

12981 ##### D.12.2.5.1.1 Payload Format

12982 The *Report Event Status* command payload shall be formatted as illustrated in Figure D-179.

Octets	**4**	**1**	**4**	**1**	**2**	**2**
Data Type	Unsigned 32-bit Integer	Unsigned 8-bit Integer	UTCTime	Unsigned 8-bit Integer	Unsigned 16-bit Integer	Unsigned 16-bit Integer
Field Name	Issuer Event ID (M)	Event Status (M)	Event Status Time (M)	Criticality Level Applied (M)	Cooling Temperature Set Point Applied (O)	Heating Temperature Set Point Applied (O)

12983

Octets	**1**	**1**	**1**
Data Type	Signed 8-bit Integer	Unsigned 8-bit Integer	8-bit BitMap
Field Name	Average Load Adjustment Percentage Applied (O)	Duty Cycle Applied (O)	Event Control (M)

12984 **Figure D-179– *Report Event Status* Command Payload**

12985 ##### D.12.2.5.1.2 Payload Details

12986 The payload shall be the same as the DRLC *Report Event Status* command (see D.2.3.3.1) with
12987 the following notes:

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

12988 **Issuer Event ID (mandatory):** Event ID specified in the *Manage Event* command or, if the
12989 command specified the current event, then the current running DRLC *Issuer Event ID*.

12990 **Event Status (mandatory)**: If the *Manage Event* command did not change the event, then this
12991 field will contain the last sent *Event Status* for the event. If the *Manage Event* command did
12992 change the DRLC event, and a new DRLC *Report Event Status* message will be created, then this
12993 field will contain the same value as the *Event Status* field of that message. If the command was to
12994 opt out of a critical event, a status of 0xF6 (Invalid Opt-out) will be returned. If the event ID does
12995 not exist, then a status of 0xF7 (Event Not Found) is returned. If the command is invalid, an APS
12996 default response would be returned.

12997 **Event Status Time (mandatory):** UTC Timestamp representing when the event status was
12998 changed. If the device does not know the current time but still remembers events, it may return the
12999 value of 0x00000000.

13000 D.12.2.5.1.3 When Generated

13001 This command is generated in response to a *Manage Event* Command.
13002

13003 ## D.12.3 Client

13004 ### D.12.3.1 Dependencies

13005 The client has no dependencies.

13006 ### D.12.3.2 Attributes

13007 The client has no attributes.

13008 ### D.12.3.3 Commands Received

13009 The client receives the cluster specific commands detailed in D.12.2.5.

13010 ### D.12.3.4 Commands Generated

13011 The client sends the cluster specific commands detailed in D.12.2.4.

13012

13013

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

D.13 MDU Pairing Cluster

Note: *The MDU Pairing Cluster description in this revision of this specification is provisionary and not certifiable. This feature set may change before reaching certifiable status in a future revision of this specification.*

D.13.1 Overview

When operating within a multi-dwelling unit (MDU), the commands within this cluster allow devices joining the NAN to acquire a list of the devices forming the 'virtual HAN' for the respective household. For details on the usage of this cluster, refer to section 5, specifically section 5.6.

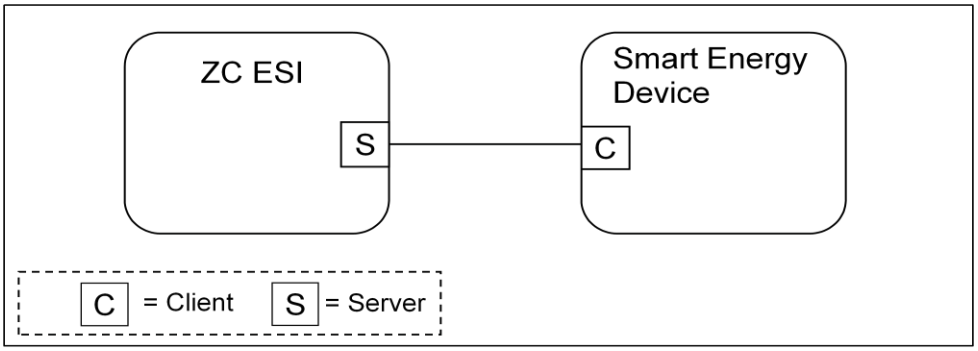


Figure D-180– MDU Pairing Cluster

D.13.2 Server

D.13.2.1 Dependencies

Support for ZCL Data Types.

D.13.2.2 Attributes

There are no attributes for the MDU Pairing cluster server.

D.13.2.3 Commands Generated

The command IDs generated by the MDU Pairing cluster server are listed in Table D-198.

Table D-198– MDU Pairing Server Commands

Command Identifier Field Value	Description	Mandatory / Optional
0x00	Pairing Response	O

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

13035 D.13.2.3.1 **Pairing Response Command**

13036 The *Pairing Response* command provides a device joining a MDU network with a list of the
13037 devices that will constitute the 'virtual HAN' for the household in which the joining device is to
13038 operate.

13039 D.13.2.3.1.1 Payload Format

Octets	**4**	**1**	**1**	**1**	**8**	**8**	**…**	**8**
Data Type	Unsigned 32-bit Integer	Unsigned 8-bit Integer	Unsigned 8-bit Integer	Unsigned 8-bit Integer	IEEE Address	IEEE Address	…	IEEE Address
Field Name	Pairing Information Version (M)	Total Number of Devices (M)	Command Index (M)	Total Number of Commands (M)	EUI64 of Device 1 (M)	EUI64 of Device 2 (O)	…	EUI64 of Device N (O)

13040 **Figure D-181– Pairing Response Command Payload**

13041 D.13.2.3.1.2 Payload Details

13042 **Pairing Information Version (mandatory):** Identifies the version of pairing information
13043 included in this command. If multiple commands are used to return this information, all
13044 commands shall contain the same value for this field. The same version shall be used to respond
13045 to all devices on the same 'virtual HAN'. Newer versions replacing older versions for the same
13046 requesting device shall use a higher value. A version of zero is an invalid value for this field.

13047 **Total Number of Devices (mandatory):** The total number of devices expected to form the
13048 'virtual HAN' (including the device to which this command is being sent). If multiple commands
13049 are used to return this information, all commands shall contain the same value for this field.

13050 **Command Index:** The *CommandIndex* is uses to count the payload fragments in the case where
13051 the entire payload does not fit into one message. The *CommandIndex* starts at 0 and is
13052 incremented for each fragment belonging to the same command.

13053 **Total Number of Commands (mandatory):** In the case where the entire payload does not fit
13054 into one message, the *Total Number of Commands* field indicates the total number of sub-
13055 commands in the message.

13056 **EUI64 of Devices:** *EUI64 of Device 1* to *EUI64 of Device N* represent the MAC address of
13057 devices that belong to the 'virtual HAN' of the requesting device; these include the requesting
13058 device itself and all other devices the requesting device shall perform service discovery and
13059 binding with. Should one message be insufficient to transfer all EUI64s, additional messages will
13060 be required (see *Command Index and Total Number of Commands* fields).

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

13061 D.13.2.3.1.3 When Generated

13062 This command is generated by a TC ESI controlling a MDU network when a *Pairing Request*
13063 command is received (see D.13.3.3.1). Receipt of a *Pairing Request* command by a TC ESI that
13064 is not operating in a MDU, or receipt by an ESI that is not the ZC/TC, shall result in a ZCL
13065 Default Response being returned with a status of UNSUP_CLUSTER_COMMAND.

13066 D.13.2.3.1.4 Effect on Receipt

13067 On receipt of this command, the requesting device shall limit the devices with which it may
13068 perform service discovery and binding to those identified in the response, instead of all devices
13069 on the ZigBee network to which it has joined. Where the entire payload of the response will not
13070 fit into a single command, the receiving device shall ensure that is successfully receives all
13071 payload fragments before the payload is used.
13072

13073 D.13.3 Client

13074 D.13.3.1 Dependencies

13075 Support for ZCL Data Types.

13076 D.13.3.2 Attributes

13077 There are no attributes for the MDU Pairing cluster client.

13078 D.13.3.3 Commands Generated

13079 The command IDs generated by the MDU Pairing cluster client are listed in Table D-199.

13080 **Table D-199– MDU Pairing Client Commands**

Command Identifier Field Value	Description	Mandatory / Optional
0x00	Pairing Request	M

13081

13082 D.13.3.3.1 **Pairing Request Command**

13083 The *Pairing Request* command allows a device joining a MDU network to determine the devices
13084 that will constitute the ‘virtual HAN’ for the household in which it is to operate.

13085 D.13.3.3.1.1 Payload Format

Octets	4	8
Data Type	Unsigned 32-bit Integer	IEEE Address

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Field Name	Local Pairing Information Version (M)	EUI64 of Requesting Device (M)

Figure D-182– Pairing Request Command Payload

D.13.3.3.1.2 Payload Details

Local Pairing Information Version (mandatory): Identifies the version of pairing information currently held on the requesting device. The TC ESI shall use this information to determine if new pairing information is to be returned. A device that does not yet have local pairing information shall set this field to zero (0x00000000).

EUI64 of Requesting Device (mandatory): Identifies the MAC address of the requesting device.

D.13.3.3.1.3 When Generated

A device should send this command to the TC ESI of a network that it is in the process of joining. It should also periodically send the command to the TC ESI in order to determine if further devices have joined its 'virtual HAN' or that the devices constituting the 'virtual HAN' have changed (*Pairing Information Version* has been updated).

D.13.3.3.1.4 Effect on Receipt

If the command is received by a TC ESI that is not operating in a MDU, or the receiving ESI is not the TC ESI, a ZCL Default Response shall be returned with a status of UNSUP_CLUSTER_COMMAND (the joining device should then follow normal (traditional) service discovery and binding procedures).

If the command contains a *Local Pairing Information Version* the matches the version of pairing information for the requesting device that is held by the TC ESI, a ZCL Default Response shall be returned with a status of WAIT_FOR_DATA (the same response should be returned if data is not yet available on the TC ESI for the requesting device).

If the command is received by a TC ESI that is controlling a MDU, the TC ESI should return a *Pairing Response* command indicating the devices that will constitute the 'virtual HAN' for the household in which the requesting device is to operate. A joining device should then follow normal (traditional) service discovery and binding procedures, but only with those devices identified in the response command.

Note that this command should not be recognized by an older Smart Energy TC ESI; this should result in that TC ESI returning a ZCL Default Response with a status of UNSUP_CLUSTER_COMMAND.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Annex E RULES AND GUIDELINES FOR OVERLAPPING EVENTS

This section describes multiple scenarios that Demand Response and Load Control devices may encounter over the Smart Energy network. The examples describe situations of overlapping events that are acceptable and where overlapping events that will be superseded due to conflicts.

E.1 Definitions

Start Time – "Start Time" field contained within the Load Control Event packet indicating when the event should start. Please note, a "Start Time" value of 0x00000000 denotes "now" and the device should use its current time as the "Start Time".

Duration – "Duration" field contained within the Load Control Event packet indicating how long the event should occur.

End Time – Time when Event completes as calculated by adding *Duration* to *Start Time*.

Scheduled Period - Represents the time between the *Start Time* and the *End Time* of the event.

Effective Start Time - Represents time at which a specific device starts a load control event based on the *Start Time* plus or minus any randomization offsets.

Effective End Time - Represents time at which a specific device ends a load control event based on the *Start Time* plus *Duration*, plus or minus any randomization offsets.

Effective Scheduled Period - Represents the time between the *Effective Start Time* and the *Effective End Time*.

Overlapping Event - Defined as an event where the *Scheduled Period* covers part or all of an existing, previously scheduled event.

Successive Events - Defined as two events where the scheduled *End Time* of the first event is equal the *Start Time* of a subsequent scheduled event.

Nested Events - Defined as two events where the scheduled *Start Time* and *End Time* of the second event falls during the *Scheduled Period* of the first scheduled event and the second event is of shorter duration than the first event.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

E.2 Rules and Guideline

The depicted behaviors and required application management decisions are driven from the following guidance and rule set:

1 Upstream Demand Response/Load Control systems and/or the ESI shall prevent mismanaged scheduling of *Overlapping Events* or *Nested Events*. It is recognized Upstream Demand Response/Load Control systems and/or the ESI will need to react to changing conditions on the grid by sending *Overlapping Events* or *Nested Events* to supersede previous directives. But those systems must have the proper auditing and management rules to prevent a cascading set of error conditions propagated by improperly scheduled events.

2 When needed, Upstream Demand Response/Load Control systems and/or the ESI may resolve any event scheduling conflicts by performing one of the following processes:

a Canceling individual events starting with the earliest scheduled event and re- issuing a new set of events.

b Canceling all scheduled events and re-issuing a new set of events.

c Sending *Overlapping Events* or *Nested Events* to supersede previous directives.

It is recommended that process 2.c is used for most situations since it can allow a smoother change between two sets of directives, but no way does it negate the responsibilities identified in rule #1.

3 When an End Device receives an event with the *End Time* in the past (*End Time* < Current Time), this event is ignored and a *Report Event Status* command is returned with the Event Status set to 0xFB (Rejected - Event was received after it had expired).

4 When an End Device receives an event with a *Start Time* in the past and an *End Time* in the future ((*Start Time* < Current Time) AND (*End Time* > Current Time)), the event is processed immediately. The Effective *Start Time* is calculated using the Current Time as the *Start Time*. Original *End Time* is preserved.

5 Regardless of the state of an event (scheduled or executing), when an *End Device* detects an *Overlapping Event* condition the latest *Overlapping Event* will take precedence over the previous event. Depending on the state of the event (scheduled or executing), one of the following steps shall take place:

a If the previous event is scheduled and not executing, the End Device returns a *Report Event Status* command (referencing the previous event) with the Event Status set to 0x07 (The event has been superseded). After the *Report Event Status* command is successfully sent, the End Device can remove the previous event schedule.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

b If the previous event is executing, the End Device shall change directly from its current state to the requested state at the *Effective Start Time* of the *Overlapping Event* (Note: Rule #4 effects *Effective Start Time*). The End Device returns a *Report Event Status* command (referencing the previous event) with the Event Status set to 0x07 (the event has been superseded).

6 Randomization ***shall not*** cause event conflicts or unmanaged gaps. To clarify:

a When event starting randomization is requested, time periods between the *Start Time* of an event and the *Effective Start Time* a device should either maintain its current state or apply changes which contribute to energy saving. Preference would be to maintain current state.

b When event ending randomization is used and the *Effective End Time* overlaps the *Effective Start Time* of a *Successive Event*, the *Effective Start Time* takes precedence. Events are not reported as superseded, End devices should report event status as it would a normal set of *Successive Events*.

c It is recommended devices apply the same Start and Stop Randomization values for consecutive events to help prevent unexpected gaps between events.

d Devices ***shall not*** artificially create a gap between *Successive Events*.

7 It is permissible to have gaps when events are not *Successive Events* or *Overlapping Events*.

8 If multiple device classes are identified for an event, future events for individual device classes (or a subset of the original event) that cause an *Overlapping Event* will supersede the original event strictly for that device class (or a subset of the original event). Note: Rule #5 applies to all *Overlapping Events*.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

E.3 Event Examples

Smart Energy devices which act upon Demand Response and Load Control events shall use the following examples for understanding and managing overlapping and superseded events. Within those examples, references to multiple device classes will be used. Figure E-1 depicts a representation of those devices in a Smart Energy network.

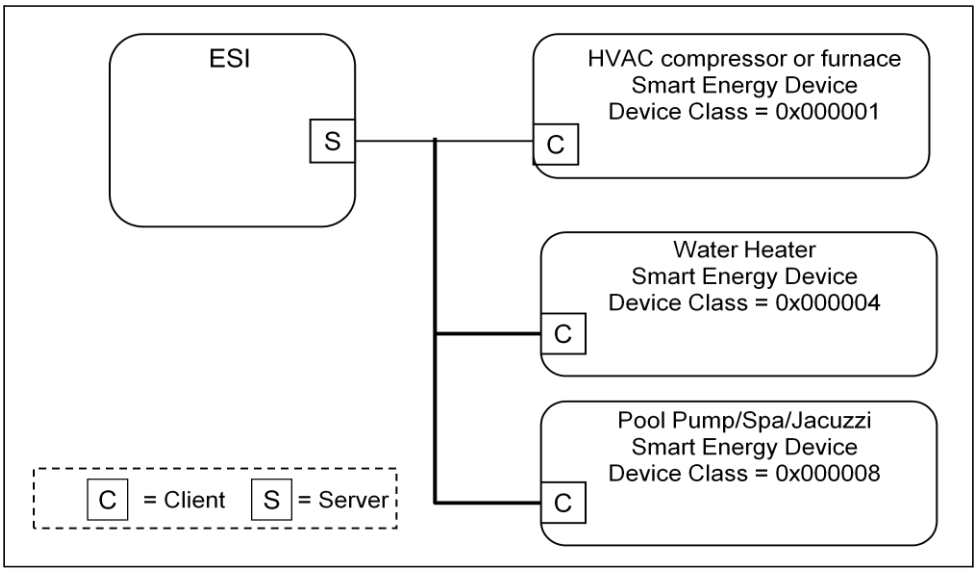


Figure E-1– Smart Energy Device Class Reference Example

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

E.3.1 Correct Overlapping Events for Different Device Classes

Figure E-2 depicts a correct series of DR/LC event for device class of 0x000001 (reference for the BitMap definition) with an event scheduled for another device class during the same period.

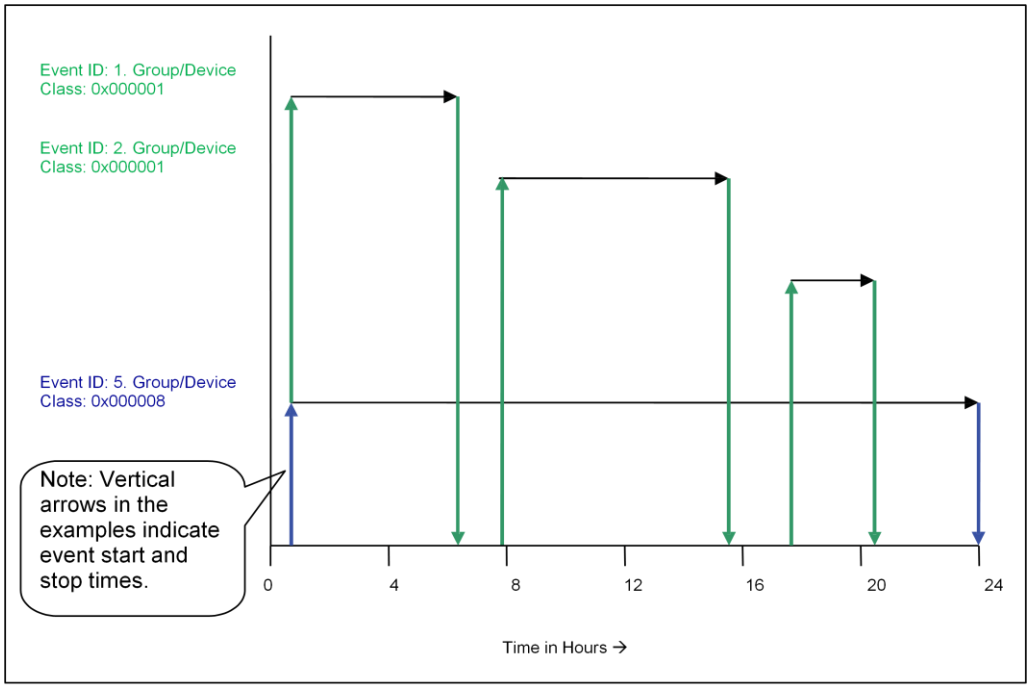


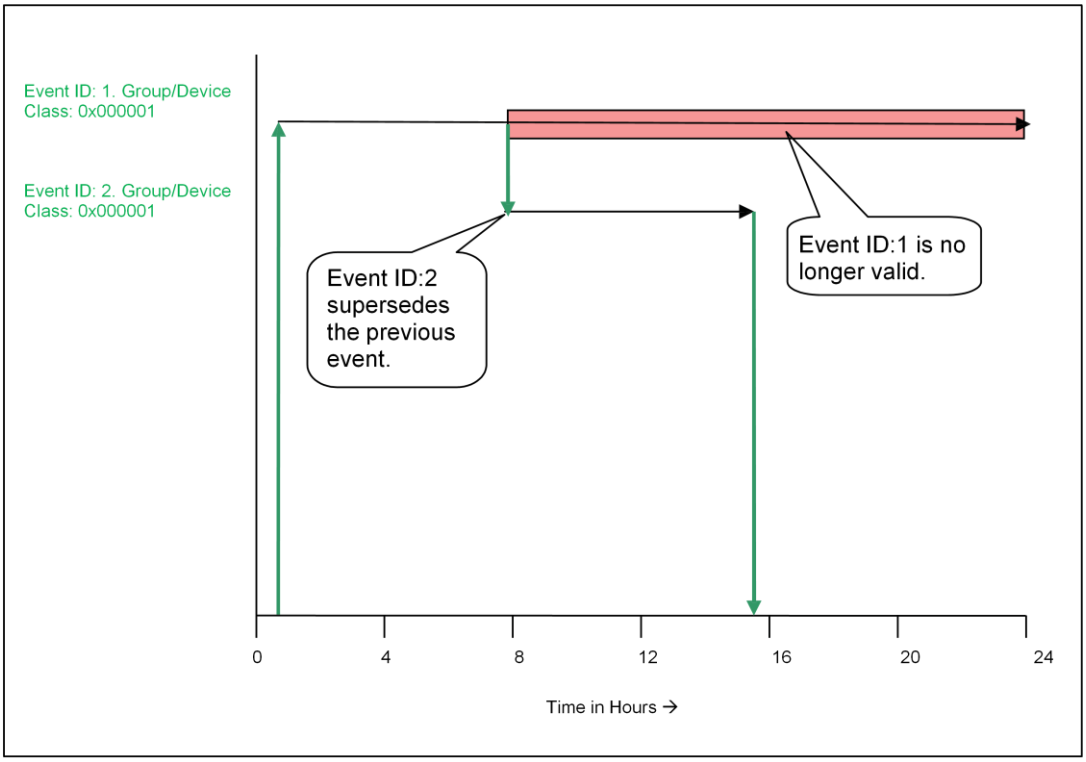
Figure E-2– Correctly Overlapping Events

In Figure E-2, Device Class 0x000001 receives a sequence of 3 unique DR/LC events to be scheduled and acted upon. During this same 24 hour period, Device Class 0x000008 receives one scheduled DR/LC event that spans across the same time period as the events scheduled for Device Class 0x0000001. Because both Device Classes are unique, there are no conflicts due to Overlapping Events.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

13219 ### E.3.2 Correct Superseded Event for a Device Class

13220 Figure E-3 below depicts a correct series of DR/LC events for device class of 0x000001
13221 (reference for the BitMap definition) where an event is scheduled then later superseded.



13222

13223 **Figure E-3– Correct Superseding of Events**

13224 In Figure E-3, Device Class 0x000001 receives DR/LC Event ID#1 setup for a 24 hour
13225 *Scheduled Period*, which later is superseded by DR/LC Event ID#2, invalidating the
13226 remainder of Event ID#1, which is cancelled.

13227

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

E.3.3 Superseding Events for Subsets of Device Classes

Figure E-4 below depicts a correct series of DR/LC events for device class of 0x000001 (reference for the BitMap definition) with an event scheduled for another device class during the same time period.

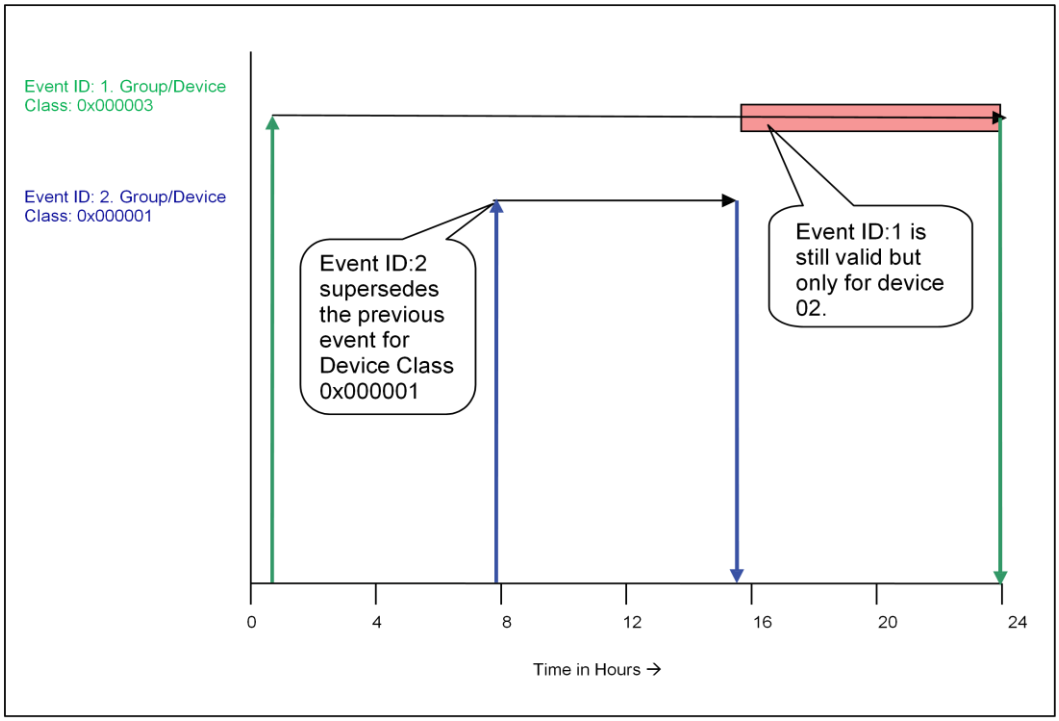


Figure E-4– Superseded Event for a Subset of Device Classes

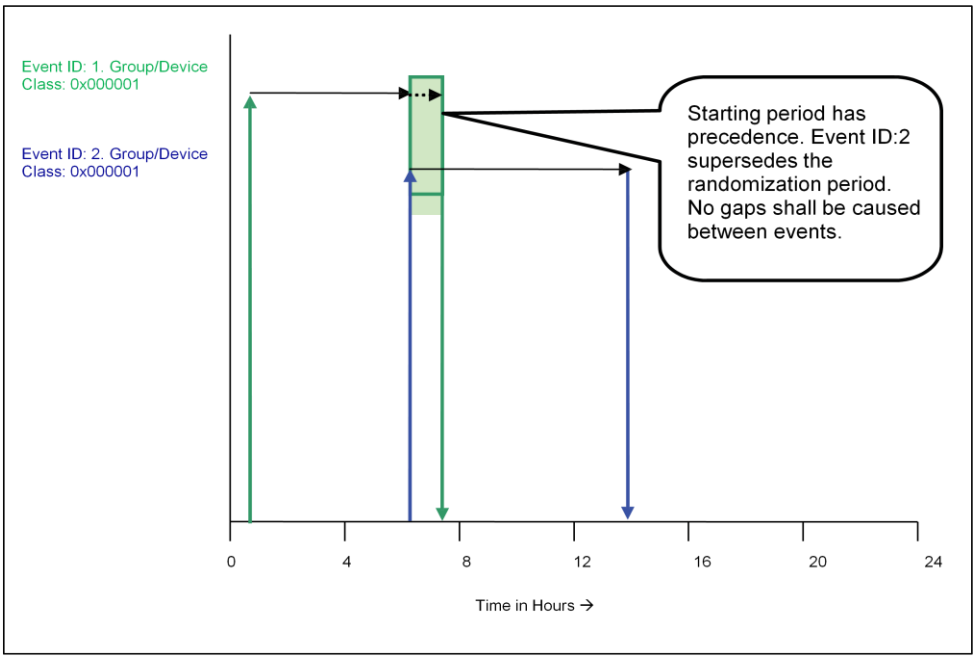
In Figure E-4, Device Class 0x000003 receives DR/LC Event ID#1 setup for a 24 hour *Scheduled Period*, which is targeted for both Device Class 0x000002 and 0x000001 (OR'ed == 0x000003). In the example, Event ID#2 is issued only for Device Class 0x000001, invalidating the remainder of Event ID#1 for that device class. DR/LC Event ID#1 is still valid for Device Class 0x000002, which in the example should run to completion.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

13240 ### E.3.4 Ending Randomization Between Events

13241 Figure E-5 below depicts an *Effective End Time* that overlaps a second scheduled DR/LC event
13242 for device class of 0x000001 (reference for the BitMap definition).

13243



13244
13245 **Figure E-5– Ending Randomization Between Events**

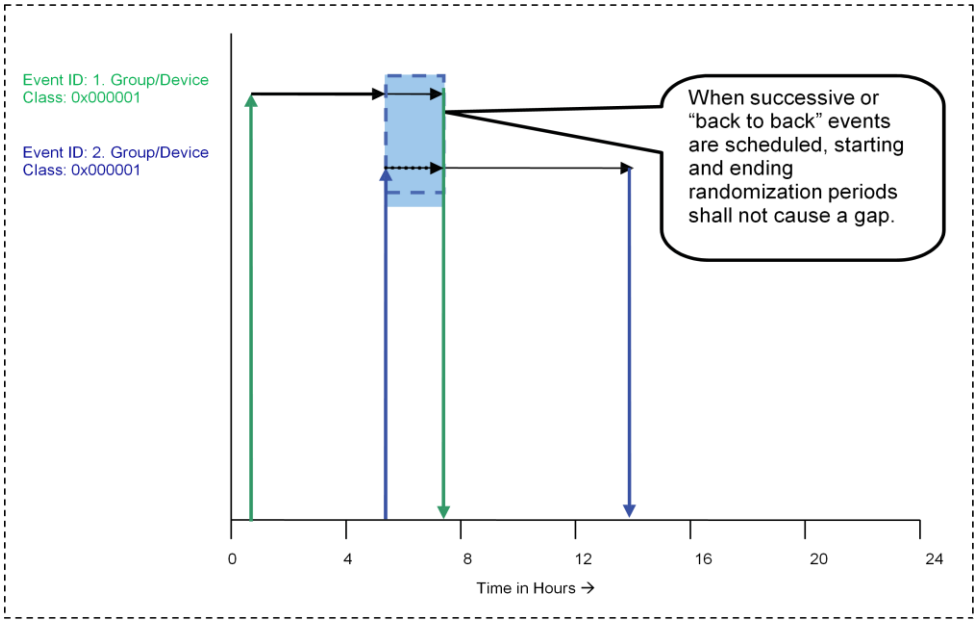
13246 In Figure E-5, Device Class 0x000001 receives a DR/LC Event ID#1 with an ending
13247 randomization setting (please refer to sub-clause D.2.2.3.1.1.1 for more detail). A second
13248 DR/LC (Event ID#2) is issued with a starting time which matches the ending time of DR/LC
13249 Event ID#1. In this situation, the *Start Time* of Event ID#2 has precedence. Event ID#1 is not
13250 reported as superseded.

13251

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

13252 ### E.3.5 Start Randomization Between Events

13253 Figure E-6 below depicts an *Effective Start Time* that overlaps a previously scheduled
13254 DR/LC event for device class of 0x000001 (reference for the BitMap definition).



13255
13256 **Figure E-6– Start Randomization Between Events**

13257 Figure E-6 above, Device Class 0x000001 receives a DR/LC Event ID#1 with an ending
13258 randomization setting (please refer to sub-clause D.2.2.3.1.1.1 for more detail). *Effective End*
13259 *Time* of Event ID#1 is not known. A second DR/LC (Event ID#2) is issued with a starting
13260 randomized setting, which has an *Effective Start Time* that could overlap or start after the
13261 *Effective End Time* of DR/LC Event ID#1. In this situation, the *Effective Start Time* of Event
13262 ID#2 has precedence but the DR/LC device must also prevent any artificial gaps caused by the
13263 *Effective Start Time* of Event ID#2 and *Effective End Time* of Event ID#1.

13264

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

E.3.6 Acceptable Gaps Caused by Start and Stop Randomization of Events

Figure E-7 below depicts an acceptable gap between two scheduled DR/LC events for device class of 0x000001 (reference for the BitMap definition) using both starting and ending randomization with both events.

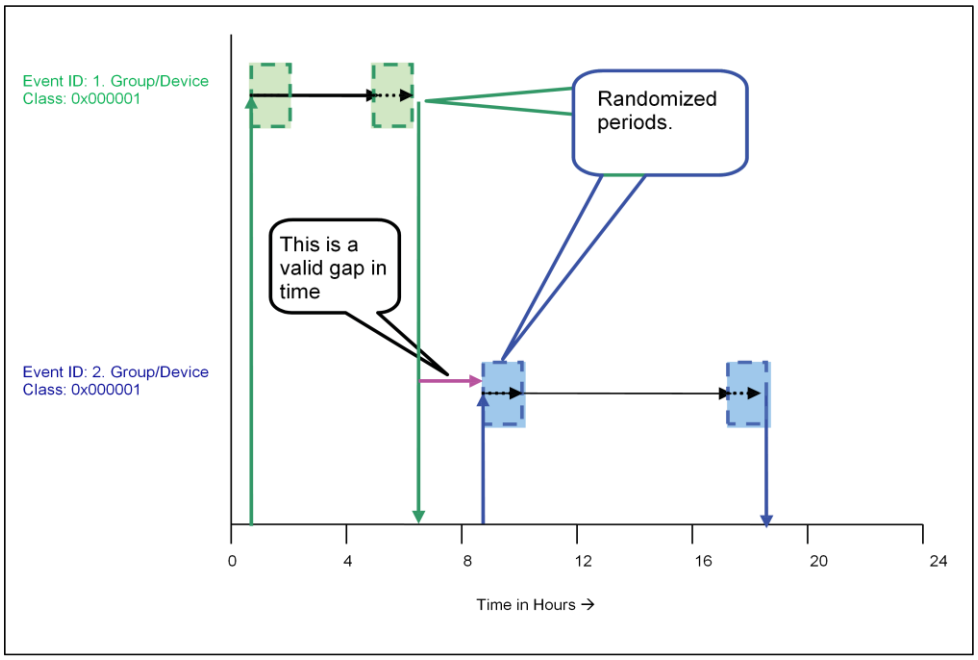


Figure E-7– Acceptable Gaps with Start and Stop Randomization

Figure E-7 above, Device Class 0x000001 receives a DR/LC Event ID#1 with both a starting and ending randomization setting (please refer to sub-clause D.2.2.3.1.1.1 for more detail). A second DR/LC Event ID#2 is also issued with both a starting and ending randomized setting. The primary configuration to note in this example is the *Effective End Time* of DR/LC Event ID#1 completes well in advance of the *Effective Start Time* of DR/LC Event ID#2. In this scenario, regardless of randomization a gap is naturally created by the scheduling of the events and is acceptable.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Annex F JOINING PROCEDURE USING PRE-CONFIGURED TRUST CENTER LINK KEYS

The secure join procedure is detailed as follows:

- The secured joining procedure is as stated in [B3] Section 4.6.3.2.3. The case used in the Smart Energy application is the "Pre-configured trust center link key and address"

- In [B3] Section 4.6.3.2.3.2, in the case of "Pre-configured Trust Center Link Key", the joining device waits for the APSME-TRANSPORT-KEY.Indication. The frame is encrypted/authenticated with the key-transport key according to the methodologies specified in sections 4.4.1.1 and 4.5.3 of the ZigBee specification r17, which describe the key-transport keys and their association with link keys, in this case the pre-configured trust center link key. The source address will be that of the Trust Center. The key transported will be the NWK Key Key type == 0x01.

- When the trust center sends the tunneled *Transport Key* command, the Extended Nonce bit on the Auxiliary Frame Header must be set to 1 on the Transport Key frame from the Trust Center to the joining child as described in [B3] Section 4.5.1. The Trust Center must also insert its long address into the Source Address field of the Auxiliary Frame Header since that information will be needed at the child to decrypt the *Transport Key* command.

- Sub-clause 5.4 of this document calls out two cases for secured join: pre- configured link keys and temporary link keys. The joining device and trust center perform the same join operation in both cases. The only difference is how the joining device and trust center treat the initial key material (either using it directly as the pre-configured link key or hashing with some data like the long address of the joining device at application level first, see Annex E for this method). From the perspective of the security joining process what happens afterwards is the secure join procedure is the same.

- In either case called out in sub-clause 5.4 of this document, the joining device is authenticated using the [B3] Section 4.6.3.2.3.2 procedure or leaves if the security timeout expires. If authenticated, the key delivered via the APSME- TRANSPORT-KEY.indication in [B3] Section 4.6.3.2.3.2 is the same for either case called out in the AMI specification sub-clause 5.4 (no matter how the application determined the pre-configured link key).

In terms of the message exchange between the child and trust center in performing the secure join procedure, the following is employed:

1 Child joining device uses NLME-JOIN.request to parent. Parent sends an APSME-UPDATE-DEVICE.request to the Trust Center on behalf of the child to the Trust Center.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

APSME-UPDATE-DEVICE.request is transported encrypted/authenticated with the NWK key that the parent has

2 Upon receipt at the trust center, the trust center must perform the following processing:

a. Validity check of the child's address to determine if a trust center link key exists between the trust center and the address provided by the joining child.

b. If the child has the trust center as its parent, the APSME-TRANSPORT- KEY.request is sent directly to the child encrypted with the key-transport key derived from the trust center link key known to the child device and the trust center, ELSE

   i. If the child does not have the trust center as its parent, the APSME- TRANSPORT-KEY command frame is encrypted using the key-transport key derived from the trust center link key shared between the child and the trust center.

c. The resulting encrypted payload is sent to the child using the APS *Tunnel* command. The APS *Tunnel* command and its (already encrypted) payload is encrypted using the NWK key from the trust center to the child's parent. On the final hop, the child's parent will perform the following processing according to [B3] Section 4.6.3.7.2:

   i. The parent sends the contents within the APS *Tunnel* command to the child without network layer encryption. The message from the parent to the joining child is an APS encrypted transport key command using the key-transport key derived from the trust center link key.

Here are the details on the message that is routed from the trust center to the joining device's parent via the *Tunnel* command:

- NWK Data Frame (Dest: Parent)
- APS Header (Command)
- APS Command Frame (Tunnel)
- Dest EUI: Child
- Tunnel Payload
- APS Header
- APS Auxiliary Header
- Encrypted Payload
- APS Command Frame (Transport Key)

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

Here are the details on the message that is routed from the joining device's parent to the joining child:

- NWK Data Frame (added by parent, Dest: child)
- APS Header (from Tunnel Payload)
- APS Auxiliary Header (from Tunnel Payload)
- Encrypted Payload (from Tunnel Payload)
- APS Command Frame (Transport Key)

The message to the child from the parent is identical if the device joins directly to the Trust Center.

As a note on the final hop contents of the payload:

- The last hop of the APME-TRANSPORT-KEY message from parent to joining child has NO network layer encryption, but does have application layer encryption
- Thus: There will be no NWK auxiliary header, but there will be an APS auxiliary header
- The APS auxiliary header will have the Key Identifier Sub-Field set to 0x02 == A key-transport key (see [B3] Section 4.5.1.1.2)
- The APS frame will be encrypted with the key-transport key derived from the pre-configured trust center link key. The pre-configured trust center link key must be part of the apsDeviceKeyPairSet in the AIB of the joining device and also known to the trust center.
- The resulting APS frame from the parent to the joining child is the APS- TRANSPORT-KEY message encrypted with the key-transport key derived from the trust center link key delivered with the key type of key-transport key (0x02).
- Per [B3] Section 4.4.3.2, the KeyType field will be set to (0x01) == Network Key
- The TransportKeyData will be the active network key and sequence number
- The joining device must set the network key and sequence number in its NWK Information Block.
- The device is then joined and authenticated.

Copyright © 2014 ZigBee Standards Organization. All rights reserved.

13371
13372
13373
13374
13375
13376
13377
13378
13379
13380
13381
13382 --- END OF DOCUMENT ---

Copyright © 2014 ZigBee Standards Organization. All rights reserved.