

1



ZigBee®
Alliance

2

3

4

5 **ZIGBEE SMART ENERGY
6 STANDARD**

7

8

9

10 **ZigBee Smart Energy Standard**

11 ZigBee Profile: 0x0109

12 Revision 19

13 Version 1.2a

14

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.

15

16

17

18

December 3, 2014

19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39

40 **This page intentionally blank**
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

58 **Notice of Use and Disclosure**

59

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

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

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

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

85 ZigBee Alliance, Inc.
86 2400 Camino Ramon, Suite 375
87 San Ramon, CA 94583, USA

88	Table of Contents	
89	Notice of Use and Disclosure.....	3
90	Table of Contents	4
91	List of Tables.....	18
92	List of Figures	24
93	Participants	30
94	Document History	31
95	1 Introduction	42
96	1.1 Scope	42
97	1.2 Purpose	42
98	1.3 Provisional Features	42
99	2 References	43
100	2.1 References	43
101	2.1.1 ZigBee Alliance Documents	43
102	2.1.2 External Reference Documents	43
103	3 Definitions.....	45
104	3.1 Conformance Levels.....	45
105	3.2 ZigBee Definitions	45
106	3.3 Smart Energy Definitions	46
107	4 Acronyms And Abbreviations.....	48
108	5 Profile Description	50
109	5.1 A ZigBee Smart Energy Network.....	50
110	5.2 ZigBee Stack Profile	53
111	5.2.1 MAC Data Polling (NLME_Requests).....	54
112	5.2.2 Application Level Queries.....	54
113	5.2.3 ZigBee Coordinator and Trust Center Recommendations	55
114	5.3 Startup Attribute Set (SAS).....	55
115	5.3.1 Startup Parameters	55
116	5.3.2 Join Parameters	57
117	5.3.3 Security Parameters	58
118	5.3.4 End Device Parameters	59
119	5.3.5 Link Status Parameters	59
120	5.3.6 Concentrator Parameters	59
121	5.3.7 APS Transport Parameters.....	59
122	5.3.8 APS Fragmentation Parameters	60
123	5.3.9 Binding Parameters	60
124	5.4 Smart Energy Profile Security	61
125	5.4.1 Joining with Preinstalled Trust Center Link Keys.....	61
126	5.4.2 Re-Joining a Secured Network.....	63
127	5.4.3 Devices Leaving the Network.....	70
128	5.4.4 Updating the Network Key	70

129	5.4.5	Updating the Link Key	71
130	5.4.6	Cluster Usage of Security Keys	73
131	5.4.7	Key Establishment Related Security Policies	75
132	5.4.8	Security Best Practices	78
133	5.5	Commissioning	81
134	5.5.1	Forming the Network (Start-up Sequence).....	82
135	5.5.2	Support for Commissioning Modes	82
136	5.5.3	Commissioning Documentation Best Practices	82
137	5.5.4	Commissioning Procedure for Different Network Types.....	83
138	5.5.5	ZigBee Smart Energy Joining, Service Discovery, and Device Binding Requirements.....	84
140	5.6	Federated Trust Center Application Guidelines	90
141	5.7	Multiple ESI Application Guidelines	91
142	5.7.1	Overview	91
143	5.7.2	Device Behavior.....	91
144	5.8	Other Smart Energy Profile Requirements and Best Practices	94
145	5.8.1	Preferred Channel Usage	94
146	5.8.2	Broadcast Policy	94
147	5.8.3	Frequency Agility	94
148	5.8.4	Key Updates	94
149	5.9	Coexistence and Interoperability with HA Devices.....	95
150	5.10	Device Descriptions.....	95
151	5.11	ZigBee Cluster Library (ZCL)	96
152	5.12	Cluster List and IDs	97
153	5.12.1	ZCL General Clusters	98
154	5.13	Coexistence with devices using other Profiles	99
155	5.13.1	Requirements for the Smart Energy Trust Center.....	100
156	5.13.2	Requirements for Smart Energy Devices	100
157	5.13.3	Requirements for Devices from other Profiles.....	101
158	6	Device Specifications	102
159	6.1	Common Clusters	102
160	6.1.1	Optional Support for Clusters with Reporting Capability.....	102
161	6.1.2	Manufacturer-Specific Clusters.....	102
162	6.1.3	Cluster Usage Restrictions	102
163	6.1.4	Identify Cluster Best Practices	103
164	6.1.5	Inter-PAN Communication	103
165	6.2	Feature and Function Description	103
166	6.3	Smart Energy Devices.....	105
167	6.3.1	Energy Service Interface.....	105
168	6.3.2	Metering Device.....	106
169	6.3.3	In-Home Display Device.....	108
170	6.3.4	Programmable Communicating Thermostat (PCT) Device.....	109
171	6.3.5	Load Control Device	109

172	6.3.6	Range Extender Device	110
173	6.3.7	Smart Appliance Device.....	110
174	6.3.8	Prepayment Terminal Device	111
175	6.3.9	Physical Device.....	112
176	6.3.10	Remote Communications Device	113
177	Annex A	Candidate ZCL Material For Use with this Profile.....	115
178	A.1	New Status Enumeration	115
179	A.2	New Attribute Reporting Status Indication	115
180	A.2.1	<i>Attribute Reporting Status</i> Attribute.....	115
181	Annex B	Enhanced Inter-PAN Transmission Mechanism.....	117
182	B.1	Scope and Purpose.....	117
183	B.2	General Description	117
184	B.2.1	What EnhancedInter-PAN Transmission Does	117
185	B.3	Service Specification	118
186	B.3.1	The INTRP-DATA.request Primitive.....	119
187	B.3.1.1	Semantics of the Service Primitive	119
188	B.3.1.2	When Generated.....	121
189	B.3.1.3	Effect on Receipt	121
190	B.3.2	The INTRP-DATA.confirm Primitive	121
191	B.3.2.1	Semantics of the Service Primitive	121
192	B.3.2.2	When Generated.....	122
193	B.3.2.3	Effect on Receipt	122
194	B.3.3	The INTRP-DATA.indication Primitive	122
195	B.3.3.1	Semantics of the Service Primitive	122
196	B.3.3.2	When Generated.....	124
197	B.3.3.3	Effect on Receipt	124
198	B.3.4	Qualifying and Testing of Inter-PAN Messages	124
199	B.4	Frame Formats.....	124
200	B.5	Frame Processing	127
201	B.5.1	Enhanced Inter-PAN Transmission.....	127
202	B.5.2	Enhanced Inter-PAN Reception.....	128
203	B.6	Initiating a Enhanced Inter-PAN Interaction.....	129
204	B.7	Best Practices	130
205	B.8	Security Requirements.....	130
206	Annex C	Key Establishment Cluster.....	131
207	C.1	Scope and Purpose.....	131
208	C.2	General Description	131
209	C.2.1	Introduction	131
210	C.2.2	Security Credentials	131
211	C.2.3	Network Security	131
212	C.2.4	Key Establishment.....	131
213	C.2.5	Symmetric Key Key Establishment	132

214	C.2.6	Public Key Key Establishment.....	133
215	C.2.7	General Exchange	133
216	C.2.7.1	Exchange Static and Ephemeral Data.....	134
217	C.2.7.2	Generate Key Bitstream.....	135
218	C.2.7.3	Derive MAC Key and Key Data	135
219	C.2.7.4	Confirm Key Using MAC	135
220	C.3	Cluster List	136
221	C.3.1	Key Establishment Cluster	136
222	C.3.1.1	Overview	136
223	C.3.1.1.1	Negotiating the Key Establishment Suite	138
224	C.3.1.2	Server.....	138
225	C.3.1.2.1	Dependencies	138
226	C.3.1.2.2	Attributes.....	138
227	C.3.1.2.3	Commands Received.....	139
228	C.3.1.2.4	Commands Generated	144
229	C.3.1.3	Client	145
230	C.3.1.3.1	Dependencies	145
231	C.3.1.3.2	Attributes.....	145
232	C.3.1.3.3	Commands Received.....	146
233	C.3.1.3.4	Commands Generated	151
234	C.4	Application Implementation	151
235	C.4.1	Network Security for Smart Energy Networks.....	151
236	C.4.2	Certificate-Based Key Establishment	152
237	C.4.2.1	Notation and Representation.....	153
238	C.4.2.1.1	Strings and String Operations	153
239	C.4.2.1.2	Integers and their Representation	153
240	C.4.2.1.3	Entities	153
241	C.4.2.2	Cryptographic Suite 1 Building Blocks.....	153
242	C.4.2.2.1	Elliptic-Curve Domain Parameters	153
243	C.4.2.2.2	Elliptic-Curve Point Representation.....	153
244	C.4.2.2.3	Elliptic-Curve Key Pair	154
245	C.4.2.2.4	ECC Implicit Certificates	154
246	C.4.2.2.5	Block-Cipher.....	155
247	C.4.2.2.6	Cryptographic Hash Function	155
248	C.4.2.2.7	Keyed Hash Function for Message Authentication.....	155
249	C.4.2.2.8	Derived Shared Secret	155
250	C.4.2.3	Cryptographic Suite 2 Building Blocks.....	155
251	C.4.2.3.1	Elliptic-Curve Point Representation.....	156
252	C.4.2.3.2	Elliptic-Curve Key Pair	156
253	C.4.2.3.3	ECC Implicit Certificates	156
254	C.4.2.3.4	Block-Cipher.....	158
255	C.4.2.3.5	Cryptographic Hash Function	158
256	C.4.2.3.6	Keyed Hash Function for Message Authentication.....	158

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

299	C.6.3.1	ECMQV Primitives	182
300	C.6.3.2	Key Derivation Function (KDF)	182
301	C.6.3.3	Initiator Transform.....	182
302	C.6.3.3.1	Ephemeral Data	182
303	C.6.3.3.2	Step Summary	183
304	C.6.3.3.3	Detailed Steps.....	183
305	C.6.3.4	Responder Transform.....	185
306	C.6.3.4.1	Ephemeral Data	185
307	C.6.3.4.2	Step Summary	185
308	C.6.3.4.3	Detailed Steps	185
309	Annex D	Smart Energy Cluster Descriptions.....	188
310	D.1	Annex Guidelines.....	188
311	D.1.1	Client/Server Model Information.....	188
312	D.1.2	Interpretation of Reserved Field Values or Bitmaps	188
313	D.2	Demand Response and Load Control Cluster.....	189
314	D.2.1	Overview	189
315	D.2.2	Server	189
316	D.2.2.1	Dependencies	189
317	D.2.2.2	Attributes	190
318	D.2.2.3	Commands Generated.....	190
319	D.2.2.3.1	Load Control Event Command	190
320	D.2.2.3.2	Cancel Load Control Event Command	196
321	D.2.2.3.3	Cancel All Load Control Events Command	198
322	D.2.2.4	Commands Received	198
323	D.2.3	Client	199
324	D.2.3.1	Dependencies	199
325	D.2.3.2	Client Cluster Attributes	199
326	D.2.3.2.1	Utility Enrollment Group Attribute	200
327	D.2.3.2.2	Start Randomization Minutes Attribute.....	200
328	D.2.3.2.3	End Randomization Minutes Attribute.....	200
329	D.2.3.2.4	DeviceClassValue Attribute	200
330	D.2.3.3	Commands Generated.....	201
331	D.2.3.3.1	Report Event Status Command.....	201
332	D.2.3.3.2	Get Scheduled Events Command	204
333	D.2.3.4	Commands Received	205
334	D.2.3.5	Attribute Reporting.....	205
335	D.2.4	Application Guidelines	205
336	D.2.4.1	Load Control Rules, Server.....	206
337	D.2.4.1.1	Load Control Server, Identifying Use of SetPoint and Offset Fields	206
338	D.2.4.1.2	Load Control Server, Editing of Scheduled Events	207
339	D.2.4.2	Load Control Rules, Client	207
340	D.2.4.2.1	Start and Stop Randomization	207
341	D.2.4.2.2	Editing of DR Control Parameters.....	207

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

384	D.4.1	Overview	350
385	D.4.2	Server	350
386	D.4.2.1	Dependencies	350
387	D.4.2.2	Attributes	350
388	D.4.2.2.1	Tier Label (Delivered) Set.....	351
389	D.4.2.2.2	Block Threshold (Delivered) Set.....	353
390	D.4.2.2.3	Block Period (Delivered) Set	356
391	D.4.2.2.4	Commodity Set.....	357
392	D.4.2.2.5	Block Price Information (Delivered) Set.....	360
393	D.4.2.2.6	Extended Price Information (Delivered) Set.....	364
394	D.4.2.2.7	Tariff Information (Delivered) Attribute Set	365
395	D.4.2.2.8	Billing Information (Delivered) Attribute Set	369
396	D.4.2.2.9	Credit Payment Attribute Set	370
397	D.4.2.2.10	Received Tier Label Attribute Set	372
398	D.4.2.2.11	Received Block Threshold Attribute Set	373
399	D.4.2.2.12	Received Block Period Attribute Set	374
400	D.4.2.2.13	Received Block Price Information Attribute Set	374
401	D.4.2.2.14	Received Extended Price Information Attribute Set.....	375
402	D.4.2.2.15	Received Tariff Information Attribute Set.....	376
403	D.4.2.2.16	Received Billing Information Attribute Set.....	377
404	D.4.2.3	Commands Received	378
405	D.4.2.3.1	Error Handling	379
406	D.4.2.3.2	GetCurrentPrice Command.....	379
407	D.4.2.3.3	GetScheduledPrices Command	380
408	D.4.2.3.4	PriceAcknowledgement Command	381
409	D.4.2.3.5	GetBlockPeriod(s) Command	382
410	D.4.2.3.6	GetConversionFactor Command.....	383
411	D.4.2.3.7	GetCalorificValue Command	384
412	D.4.2.3.8	GetTariffInformation Command.....	385
413	D.4.2.3.9	GetPriceMatrix Command	386
414	D.4.2.3.10	GetBlockThresholds Command.....	387
415	D.4.2.3.11	GetCO ₂ Value Command.....	387
416	D.4.2.3.12	GetTierLabels Command.....	388
417	D.4.2.3.13	GetBillingPeriod Command.....	389
418	D.4.2.3.14	GetConsolidatedBill Command	390
419	D.4.2.3.15	CPPEventResponse Command	391
420	D.4.2.3.16	GetCreditPayment Command	392
421	D.4.2.3.17	GetCurrencyConversion Command.....	392
422	D.4.2.3.18	GetTariffCancellation Command.....	393
423	D.4.2.4	Commands Generated.....	393
424	D.4.2.4.1	Publish Price Command.....	394
425	D.4.2.4.2	Publish Block Period Command	405
426	D.4.2.4.3	PublishConversionFactor Command.....	409

427	D.4.2.4.4	PublishCalorificValue Command	410
428	D.4.2.4.5	PublishTariffInformation Command	410
429	D.4.2.4.6	PublishPriceMatrix Command.....	413
430	D.4.2.4.7	PublishBlockThresholds Command.....	416
431	D.4.2.4.8	PublishCO ₂ Value Command	418
432	D.4.2.4.9	PublishTierLabels Command.....	420
433	D.4.2.4.10	PublishBillingPeriod Command	421
434	D.4.2.4.11	PublishConsolidatedBill Command.....	422
435	D.4.2.4.12	PublishCPPEvent Command	424
436	D.4.2.4.13	PublishCreditPayment Command	427
437	D.4.2.4.14	PublishCurrencyConversion Command.....	428
438	D.4.2.4.15	CancelTariff Command	429
439	D.4.3	Client	430
440	D.4.3.1	Dependencies.....	430
441	D.4.3.2	Attributes	431
442	D.4.3.2.1	PriceIncreaseRandomizeMinutes Attribute	431
443	D.4.3.2.2	PriceDecreaseRandomizeMinutes Attribute.....	432
444	D.4.3.2.3	CommodityType Attribute	432
445	D.4.3.3	Commands Received	432
446	D.4.3.4	Commands Generated.....	432
447	D.4.4	Application Guidelines	432
448	D.4.4.1	Registering for Commands	432
449	D.4.4.2	Attribute Reporting.....	432
450	D.4.4.3	Block Tariffs.....	433
451	D.4.4.3.1	TOU Charging Only	433
452	D.4.4.3.2	Block Charging only	433
453	D.4.4.3.3	Block/TOU Combination Charging	434
454	D.4.4.3.4	Application Guidelines for Block Pricing under specific events	434
455	D.4.4.4	Handling of Enhanced Tariffs	435
456	D.4.4.4.1	Block Charging	435
457	D.4.4.4.2	TOU Charging	436
458	D.4.4.4.3	TOU/Block Charging.....	436
459	D.4.4.4.4	Critical Peak Pricing	436
460	D.4.4.4.5	Generation Charging	437
461	D.5	Messaging Cluster.....	438
462	D.5.1	Overview	438
463	D.5.2	Server	438
464	D.5.2.1	Dependencies.....	438
465	D.5.2.2	Attributes	438
466	D.5.2.3	Commands Generated.....	438
467	D.5.2.3.1	Display Message Command.....	439
468	D.5.2.3.2	Cancel Message Command	442

469	D.5.2.3.3	Display Protected Message Command.....	442
470	D.5.2.3.4	Cancel All Messages Command	443
471	D.5.3	Client	443
472	D.5.3.1	Dependencies.....	443
473	D.5.3.2	Attributes	443
474	D.5.3.3	Commands Generated.....	443
475	D.5.3.3.1	Get Last Message Command.....	444
476	D.5.3.3.2	Message Confirmation Command.....	444
477	D.5.3.3.3	GetMessageCancellation Command.....	445
478	D.5.4	Application Guidelines	446
479	D.6	Tunneling Cluster.....	447
480	D.6.1	Overview	447
481	D.6.2	Server	450
482	D.6.2.1	Dependencies.....	450
483	D.6.2.2	Attributes	450
484	D.6.2.2.1	CloseTunnelTimeout Attribute.....	451
485	D.6.2.3	Parameters	451
486	D.6.2.4	Commands Received	451
487	D.6.2.4.1	RequestTunnel Command	452
488	D.6.2.4.2	CloseTunnel Command.....	453
489	D.6.2.4.3	TransferData Command	454
490	D.6.2.4.4	TransferDataError Command.....	455
491	D.6.2.4.5	AckTransferData Command.....	456
492	D.6.2.4.6	ReadyData Command	457
493	D.6.2.4.7	Get Supported Tunnel Protocols Command	458
494	D.6.2.5	Commands Generated.....	459
495	D.6.2.5.1	RequestTunnelResponse Command.....	459
496	D.6.2.5.2	TransferData Command	460
497	D.6.2.5.3	TransferDataError Command.....	461
498	D.6.2.5.4	AckTransferData Command.....	461
499	D.6.2.5.5	ReadyData Command	461
500	D.6.2.5.6	Supported Tunnel Protocols Response Command	461
501	D.6.2.5.7	TunnelClosureNotification Command.....	463
502	D.6.3	Client	463
503	D.6.3.1	Dependencies.....	463
504	D.6.3.2	Attributes	463
505	D.6.3.3	Commands Received	463
506	D.6.3.4	Commands Generated.....	463
507	D.7	Prepayment Cluster	465
508	D.7.1	Overview	465
509	D.7.2	Server	466
510	D.7.2.1	Dependencies.....	466
511	D.7.2.2	Attributes	466

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

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

597	D.10.2.4.2 Publish Change of Supplier Command.....	556
598	D.10.2.4.3 Request New Password Response Command.....	558
599	D.10.2.4.4 Update SiteID Command.....	559
600	D.10.2.4.5 SetEventConfiguration Command	560
601	D.10.2.4.6 GetEventConfiguration Command	562
602	D.10.2.4.7 Update CIN Command	562
603	D.10.3 Client	564
604	D.10.3.1 Dependencies.....	564
605	D.10.3.2 Attributes	564
606	D.10.3.2.1 Supplier Attribute Set	564
607	D.10.3.2.2 Price Event Configuration Attribute Set.....	565
608	D.10.3.2.3 Metering Event Configuration Attribute Set.....	569
609	D.10.3.2.4 Messaging Event Configuration Attribute Set.....	578
610	D.10.3.2.5 Prepayment Event Configuration Attribute Set	579
611	D.10.3.2.6 Calendar Event Configuration Attribute Set.....	583
612	D.10.3.2.7 Device Management Event Configuration Attribute Set	585
613	D.10.3.2.8 Tunnel Event Configuration Attribute Set.....	587
614	D.10.3.2.9 OTA Event Configuration Attribute Set	589
615	D.10.3.3 Commands Received	591
616	D.10.3.4 Commands Generated.....	591
617	D.10.4 Application Guidelines	591
618	D.10.4.1 Passwords	591
619	D.10.4.2 Consumer Password Use Case	591
620	D.10.4.3 Engineer Password Use Case	591
621	D.10.4.4 Password Security Recommendations.....	591
622	D.11 Events Cluster	592
623	D.11.1 Overview	592
624	D.11.2 Server.....	593
625	D.11.2.1 Dependencies.....	593
626	D.11.2.2 Attributes	594
627	D.11.2.3 Commands Received	594
628	D.11.2.3.1 Get Event Log Command	594
629	D.11.2.3.2 Clear Event Log Request Command.....	596
630	D.11.2.4 Commands Generated.....	597
631	D.11.2.4.1 Publish Event Command.....	597
632	D.11.2.4.2 Publish Event Log Command	598
633	D.11.2.4.3 Clear Event Log Response Command	600
634	D.11.3 Client	601
635	D.11.3.1 Dependencies.....	601
636	D.11.3.2 Attributes	601
637	D.11.3.3 Commands Received	601
638	D.11.3.4 Commands Generated.....	601

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

List of Tables

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 – <i>Attribute Reporting Status</i> Attribute	115
711	Table A-3 – Attribute Reporting Status Enumerations	116
712	Table B-1 – Parameters of the INTRP-DATA. <i>request</i>	120
713	Table B-2 – Parameters of the INTRP-DATA. <i>confirm</i>	121
714	Table B-3 – Parameters of the INTRP-DATA. <i>indication</i>	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 <i>KeyEstablishmentSuite</i> Attribute	139
719	Table C-5 – Received Command IDs for the Key Establishment Cluster Server	139
720	Table C-6 – <i>Terminate Key Establishment</i> Command Status Field	143
721	Table C-7 – Key Establishment Attribute Sets	145

722	Table C-8– Attributes of the Information Attribute Set.....	145
723	Table C-9– Values of the <i>KeyEstablishmentSuite</i> Attribute	146
724	Table C-10– Received Command IDs for the Key Establishment Cluster Client	146
725	Table C-11– <i>Terminate Key Establishment</i> Command Status Field.....	150
726	Table C-12– Values of the <i>KeyUsage</i> Field.....	157
727	Table C-13– ECC Implicit Certificate format.....	157
728	Table C-14– Parameters Used by Methods of the CBKE Protocol	159
729	Table D-1– Command IDs for the Demand Response and Load Control Server	190
730	Table D-2– Device Class Field BitMap/Encoding	191
731	Table D-3– Criticality Levels.....	192
732	Table D-4– Event Control Field BitMap.....	195
733	Table D-5– Cancel Control	197
734	Table D-6– Cancel All Command Cancel Control Field	198
735	Table D-7– Demand Response Client Cluster Attributes.....	199
736	Table D-8– Generated Command IDs for the Demand Response and Load Control Client.....	201
737	Table D-9– Event Status Field Values	202
738	Table D-10– Metering Cluster Server Attribute Sets.....	214
739	Table D-11– Reading Information Attribute Set.....	215
740	Table D-12– Block Enumerations	220
741	Table D-13– Supply Status Attribute Enumerations	221
742	Table D-14– TOU Information Attribute Set.....	224
743	Table D-15– Meter Status Attribute Set.....	228
744	Table D-16– Mapping of the <i>Status</i> Attribute (Electricity).....	229
745	Table D-17– Meter Status Attribute (Gas)	229
746	Table D-18– Meter Status Attribute (Water)	230
747	Table D-19– Meter Status Attribute (Heat and Cooling).....	230
748	Table D-20– General Flags of the Extended Status BitMap	232
749	Table D-21– Electricity -Meter specific Flags of the Extended Status BitMap.....	233
750	Table D-22– Gas-Meter specific Flags of the Extended Status BitMap	234
751	Table D-23– LowMediumHighStatus Attribute	235
752	Table D-24– Formatting Examples.....	236
753	Table D-25– Formatting Attribute Set	236
754	Table D-26– <i>UnitofMeasure</i> Attribute Enumerations	238
755	Table D-27– <i>MeteringDeviceType</i> Attribute.....	242
756	Table D-28– <i>TemperatureUnitOfMeasure</i> Enumeration	244
757	Table D-29– Historical Attribute Set	248
758	Table D-30– Load Profile Configuration Attribute Set	258
759	Table D-31– Supply Limit Attribute Set.....	259
760	Table D-32– Block Information Attribute Set (Delivered)	262
761	Table D-33– Alarm Attribute Set	266
762	Table D-34– Alarm Code Groups	267
763	Table D-35– Generic Alarm Group.....	268

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

805	Table D-77– <i>Commodity</i> Attribute Set	357
806	Table D-78– Values and Descriptions for the <i>CalorificValueUnit</i> Attribute	359
807	Table D-79– Block Price Information Attribute Set	360
808	Table D-80– Extended Price Information Set (TOU charging only)	364
809	Table D-81– Tariff Information Attribute Set.....	365
810	Table D-82– TierBlockMode Enumeration	366
811	Table D-83– CO ₂ Unit Enumeration	368
812	Table D-84– Billing Information Attribute Set.....	369
813	Table D-85– Credit Payment Attribute Set.....	370
814	Table D-86– CreditPaymentStatus Enumeration	371
815	Table D-87– PaymentDiscountDuration Enumerations.....	372
816	Table D-88– Received Tier Label Attribute Set	372
817	Table D-89– Received Block Threshold Attribute Set	373
818	Table D-90– Received Block Period Attribute Set	374
819	Table D-91– Received Block Price Attribute Set	374
820	Table D-92– Received Extended Price Information Attribute Set (TOU charging only).....	375
821	Table D-93– Received Tariff Information Attribute Set.....	376
822	Table D-94– Received Billing Information Attribute Set.....	377
823	Table D-95– Received Command IDs for the Price Cluster	378
824	Table D-96– Generated Command IDs for the Price Cluster.....	393
825	Table D-97– Price Tier Sub-field Enumerations	396
826	Table D-98– Register Tier Sub-field Enumerations.....	398
827	Table D-99– Alternate Cost Unit Enumerations	399
828	Table D-100– Price Control Field BitMap	400
829	Table D-101– Generation Tier Enumerations	400
830	Table D-102– Extended Price Tier Field Enumerations	403
831	Table D-103– Extended Register Tier Field Enumerations	404
832	Table D-104– Block Period Control Field BitMap	407
833	Table D-105– Block Period DurationTimebase Enumeration	407
834	Table D-106– Block Period Duration Control Enumeration	408
835	Table D-107– Tariff Resolution Period Enumeration.....	408
836	Table D-108– Tariff Type Enumeration	412
837	Table D-109– Tariff Charging Scheme Enumeration	412
838	Table D-110– PublishPriceMatrix Sub-Payload Control Bitmap	415
839	Table D-111– PublishBlockThresholds Sub-Payload Control Bitmap.....	417
840	Table D-112– CPP Auth Enumeration.....	425
841	Table D-113– CPP Auth Enumeration.....	425
842	Table D-114– Currency Change Control	429
843	Table D-115– Price Client Cluster Attributes	431
844	Table D-116– Generated Command IDs for the Messaging Server.....	439
845	Table D-117– Message Control Field Bit Map	439
846	Table D-118– Extended Message Control Field Bit Map	441

847	Table D-119– Messaging Client Commands	443
848	Table D-120– Message Confirmation Control.....	445
849	Table D-121– Tunneling Server Attributes.....	450
850	Table D-122– Cluster Parameters Passed Through Commands	451
851	Table D-123– Cluster -specific Commands Received by the Server	452
852	Table D-124– ProtocolID Enumerations.....	452
853	Table D-125– TransferDataStatus Values	456
854	Table D-126– Cluster-Specific Commands Sent by the Server	459
855	Table D-127– TunnelStatus Values	460
856	Table D-128– Prepayment Attribute Sets.....	466
857	Table D-129– <i>Prepayment Information</i> Attribute Set.....	467
858	Table D-130– <i>Payment Control Configuration</i> Attribute	468
859	Table D-131– <i>Credit Status</i> Attribute	471
860	Table D-132– OverallDebtCap Example	471
861	Table D-133– Top-up Attribute Set	474
862	Table D-134– Debt Attribute Set	477
863	Table D-135– Debt Recovery Method Enumerations.....	479
864	Table D-136– <i>Recovery Frequency</i> Field Enumerations	479
865	Table D-137– Alarm Attribute Set.....	480
866	Table D-138– Prepayment Alarm Status Indicators	481
867	Table D-139– Alarms Code Group	482
868	Table D-140– PrepayGenericAlarmGroup	482
869	Table D-141– PrepaySwitchAlarmGroup	482
870	Table D-142– PrepayEventAlarmGroup.....	483
871	Table D-143– Historical Cost Consumption Information Attribute Set	483
872	Table D-144– <i>CurrencyScalingFactor</i> Enumerations.....	490
873	Table D-145– Cluster -specific Commands Received by the Server	493
874	Table D-146– Originating Device Field Enumerations	494
875	Table D-147– Debt Amount Type Field Enumerations	496
876	Table D-148– Credit Type Field Enumerations	499
877	Table D-149– Debt Type Field Enumerations	504
878	Table D-150– Cluster -specific Commands Sent by the Server.....	506
879	Table D-151– Snapshot Payload Cause	508
880	Table D-152– Snapshot Payload Type.....	509
881	Table D-153– Friendly Credit BitMap.....	510
882	Table D-154– Result Type Field Enumerations.....	512
883	Table D-155–Calendar Data Structures	522
884	Table D-156–Calendar Cluster Attribute Sets	524
885	Table D-157– Auxiliary Switch Label Attribute Set	524
886	Table D-158– Commands Generated by the Calendar Cluster Server.....	525
887	Table D-159– Calendar Type Enumeration	527

888	Table D-160– Calendar Time Reference Enumeration.....	527
889	Table D-161– Commands Received by the Calendar Cluster Server	537
890	Table D-162– Device Management Attribute Sets	545
891	Table D-163– Supplier Control Attribute Set	545
892	Table D-164– Proposed Change Control	547
893	Table D-165– Contactor State Bit Combinations	548
894	Table D-166– Tenancy Control Attribute Set	549
895	Table D-167– Backhaul Control Attribute Set.....	550
896	Table D-168– State of the WAN Connection	550
897	Table D-169– HAN Control Attribute Set	550
898	Table D-170– Cluster Specific Commands Received by the Server	551
899	Table D-171– Cluster Specific Commands Generated by the Server	554
900	Table D-172– Password Type Enumeration	558
901	Table D-173– Configuration Control Enumeration	560
902	Table D-174– Device Management Client Attribute Sets	564
903	Table D-175– Supplier Attribute Sets.....	564
904	Table D-176– Price Event Configuration Attribute Set	565
905	Table D-177– Event Configuration Bitmaps	569
906	Table D-178– Metering Event Configuration Attribute Set.....	569
907	Table D-179– Messaging Event Configuration Attribute Set	578
908	Table D-180– Prepayment Event Configuration Attribute Set	579
909	Table D-181– Calendar Event Configuration Attribute Set.....	584
910	Table D-182– Device Management Event Configuration Attribute Set	585
911	Table D-183– Tunneling Event Configuration Attribute Set.....	587
912	Table D-184– OTA Event Configuration Attribute Set	589
913	Table D-185– Cluster Specific Commands Received by the Server	594
914	Table D-186– Log ID Enumeration	595
915	Table D-187– Event Control Bitmap	595
916	Table D-188– Cluster Specific Commands Generated by the Server	597
917	Table D-189– Event Action Control Bitmap	598
918	Table D-190– Log Payload Control Bitmap	599
919	Table D-191– ClearedEventsLogs Bitmap	600
920	Table D-192– Energy Management Server Cluster Attributes.....	602
921	Table D-193– Load Control State BitMap/Encoding	603
922	Table D-194– Current Event Status BitMap/Encoding	605
923	Table D-195– Received Command IDs for the Energy Management Server	607
924	Table D-196– Action(s) Required BitMap/Encoding	608
925	Table D-197– Generated Command IDs for the Energy Management Server	609
926	Table D-198– MDU Pairing Server Commands	611
927	Table D-199– MDU Pairing Client Commands.....	613
928		

929	List of Figures	
930	Figure 5-1 Utility Private HAN.....	51
931	Figure 5-2 Utility Private NAN.....	52
932	Figure 5-3 Customer Private NAN.....	52
933	Figure 5-4 An Example of a MDU Setup	53
934	Figure 5-5 Successful Join and Registration	72
935	Figure 5-6 Node Communication with Other Nodes on the Network Using APS Layer Encryption	77
937	Figure 5-7 Smart Energy Device Installation Code Process	78
938	Figure 5-8 Installation Code Use with the Trust Center	79
939	Figure B-1– ZigBee Stack with Stub APS	118
940	Figure B-2– Normal ZigBee Frame	124
941	Figure B-3– Enhanced Inter-PAN ZigBee Frame	125
942	Figure B-4– Stub NWK Header Format	125
943	Figure B-5– NWK Frame Control Field	125
944	Figure B-6– Stub APS Header Format.....	125
945	Figure B-7– APS Frame Control Field.....	126
946	Figure C-1– Overview of General Exchange	134
947	Figure C-2– Typical Usage of the Key Establishment Cluster	136
948	Figure C-3– Key Establishment Command Exchange.....	137
949	Figure C-4– <i>Initiate Key Establishment Request</i> Command Payload.....	140
950	Figure C-5– <i>Ephemeral Data Request</i> Command Payload.....	142
951	Figure C-6– <i>Confirm Key Request</i> Command Payload.....	142
952	Figure C-7– <i>Terminate Key Establishment</i> Command Payload.....	143
953	Figure C-8– <i>Initiate Key Establishment Response</i>	147
954	Figure C-9– <i>Ephemeral Data Response</i> Command Payload	148
955	Figure C-10– <i>Confirm Key Response</i> Command Payload	149
956	Figure C-11– <i>Terminate Key Establishment</i> Command Payload.....	150
957	Figure C-12– Key Establishment Command Exchange.....	165
958	Figure C-13– Key Establishment Command Exchange.....	177
959	Figure D-1– Demand Response/Load Control Cluster Client Server Example.....	189
960	Figure D-2– <i>Load Control Event</i> Command Payload.....	190
961	Figure D-3– <i>Cancel Load Control Event</i> Payload	196
962	Figure D-4– <i>Cancel All Load Control Events</i> Command Payload	198
963	Figure D-5– <i>Report Event Status</i> Command Payload	202
964	Figure D-6– <i>Get Scheduled Events</i> Command Payload	205
965	Figure D-7– Example of Both a Successful and an Overridden Load.....	209
966	Figure D-8– Example of a Load Curtailment Superseded and.....	210
967	Figure D-9– Standalone ESI Model with Mains Powered Metering Device	211
968	Figure D-10– Standalone ESI Model with Battery Powered Metering Device	212

969	Figure D-11– ESI Model with Integrated Metering Device.....	213
970	Figure D-12– <i>Get Profile Response</i> Command Payload	289
971	Figure D-13– <i>Request Fast Poll Mode Response</i> Command Payload.....	292
972	Figure D-14– <i>ScheduleSnapshotResponse</i> Command Payload	293
973	Figure D-15– Snapshot Response Payload Sub-Payload.....	293
974	Figure D-16– <i>TakeSnapshotResponse</i> Command Payload	294
975	Figure D-17– <i>Publish Snapshot</i> 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 <i>GetSampledDataResponse</i> Command Payload	306
986	Figure D-28– Format of the <i>ConfigureMirror</i> Command Payload.....	307
987	Figure D-29– <i>MirrorReportAttributeResponse</i> Command Enabled	308
988	Figure D-30– <i>MirrorReportAttributeResponse</i> Command Disabled	309
989	Figure D-31– <i>NotificationScheme</i> Enumerations.....	310
990	Figure D-32– <i>ConfigureNotificationScheme</i> Command Payload	311
991	Figure D-33– <i>ConfigureNotificationFlags</i> Command Payload	312
992	Figure D-34– <i>Bit Field Allocation</i> Command Sub Payload.....	313
993	Figure D-35– <i>GetNotifiedMessage</i> Command Payload	314
994	Figure D-36– Supply Status Response Command Payload	315
995	Figure D-37– <i>StartSamplingResponse</i> Command Payload.....	316
996	Figure D-38– <i>Get Profile</i> Command Payload.....	323
997	Figure D-39– <i>Request Mirror Response</i> Command Payload	324
998	Figure D-40– <i>Mirror Removed</i> Command Payload	325
999	Figure D-41– <i>Request Fast Poll Mode</i> Command Payload	325
1000	Figure D-42– <i>ScheduleSnapshot</i> Command Payload.....	326
1001	Figure D-43– <i>SnapshotSchedulePayload</i> Format	326
1002	Figure D-44– <i>TakeSnapshot</i> Command Payload	327
1003	Figure D-45– <i>GetSnapshot</i> Command Payload	328
1004	Figure D-46– <i>StartSampling</i> Command Payload.....	329
1005	Figure D-47– <i>GetSampledData</i> Command Payload	330
1006	Figure D-48– <i>MirrorReportAttributeResponse</i> Command Payload	331
1007	Figure D-49– <i>ResetLoadLimitCounter</i> 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– <i>SetSupplyStatus</i> Command Payload	336
1011	Figure D-53– <i>SetUncontrolledFlowThreshold</i> Command Payload	337

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– <i>GetCurrentPrice</i> Command Payload.....	379
1017	Figure D-59– <i>Get Current Price</i> Command Options Field.....	380
1018	Figure D-60– <i>GetScheduledPrices</i> Command Payload	381
1019	Figure D-61– <i>PriceAcknowledgement</i> Command Payload	381
1020	Figure D-62– <i>Get Block Period(s)</i> Command Payload.....	382
1021	Figure D-63– <i>GetConversionFactor</i> Command Payload.....	383
1022	Figure D-64– <i>GetCalorificValue</i> Command Payload	384
1023	Figure D-65– <i>GetTariffInformation</i> Command Payload	385
1024	Figure D-66– <i>GetPriceMatrix</i> Command Payload	386
1025	Figure D-67– <i>GetBlockThresholds</i> Command Payload.....	387
1026	Figure D-68– <i>GetCO₂Value</i> Command Payload	387
1027	Figure D-69– <i>GetTierLabels</i> Command Payload.....	388
1028	Figure D-70– <i>GetBillingPeriod</i> Command Payload	389
1029	Figure D-71– <i>GetConsolidatedBill</i> Command Payload.....	390
1030	Figure D-72– <i>CPPEventResponse</i> Command Payload	391
1031	Figure D-73– <i>GetCreditPayment</i> Command Payload	392
1032	Figure D-74– <i>Publish Price</i> Command Payload	395
1033	Figure D-75– <i>Publish Block Period</i> Command Payload.....	406
1034	Figure D-76– <i>PublishConversionFactor</i> Command Payload	409
1035	Figure D-77– <i>PublishCalorificValue</i> Command Payload.....	410
1036	Figure D-78– <i>PublishTariffInformation</i> Command Payload	411
1037	Figure D-79– <i>PublishPriceMatrix</i> Command Payload	414
1038	Figure D-80– <i>PriceMatrix</i> Command Sub-Payload.....	415
1039	Figure D-81– <i>PublishBlockThresholds</i> Command Payload.....	416
1040	Figure D-82– <i>BlockThreshold</i> Sub-Payload Format	417
1041	Figure D-83– <i>PublishCO₂Value</i> Command Payload	419
1042	Figure D-84– <i>PublishTierLabels</i> Command Payload	420
1043	Figure D-85– <i>PublishBillingPeriod</i> Command Payload	421
1044	Figure D-86– <i>PublishConsolidatedBill</i> Command Payload	423
1045	Figure D-87– <i>PublishCPPEvent</i> Command Payload	424
1046	Figure D-88– CPP Event Flow.....	426
1047	Figure D-89– <i>PublishCreditPayment</i> Command Payload	427
1048	Figure D-90– <i>PublishCurrencyConversion</i> Command Payload.....	428
1049	Figure D-91– <i>CancelTariff</i> Command Payload	430
1050	Figure D-92– Messaging Cluster Client/Server Example	438
1051	Figure D-93– <i>Display Message</i> Command Payload	439
1052	Figure D-94– <i>Cancel Message</i> Command Payload	442
1053	Figure D-95– <i>Cancel All Messages</i> Command Payload.....	443

1054	Figure D-96– <i>Message Confirmation</i> Command Payload	444
1055	Figure D-97– <i>GetMessageCancellation</i> Command Payload	445
1056	Figure D-98– Client/Server Message Command Exchanges	446
1057	Figure D-99– A Client Requests a Tunnel From a Server to Exchange Complex	447
1058	Figure D-100– SE Device 1 (Client) Requests a Tunnel From SE Device 2 (Server) to Transfer Data Without Flow Control (Default)	449
1059	Figure D-101– SE Device 1 (Client) Requests a Tunnel From SE Device 2 (Server) to Transfer Data With Flow Control	450
1062	Figure D-102– <i>RequestTunnel</i> Command Payload	452
1063	Figure D-103– <i>CloseTunnel</i> Command Payload.....	454
1064	Figure D-104– <i>TransferData</i> Command Payload	454
1065	Figure D-105– <i>TransferDataError</i> Command Payload	455
1066	Figure D-106– <i>AckTransferData</i> Command Payload	456
1067	Figure D-107– <i>ReadyData</i> Command Payload.....	457
1068	Figure D-108– <i>Get Supported Tunnel Protocols</i> Command Payload	458
1069	Figure D-109– <i>RequestTunnelResponse</i> Command Payload	459
1070	Figure D-110– <i>TransferData</i> Command Payload	461
1071	Figure D-111– <i>Supported Tunnel Protocols Response</i> Command Payload.....	462
1072	Figure D-112– <i>Supported Tunnel Protocols Response</i> Command Protocol Fields	462
1073	Figure D-113– <i>TunnelClosureNotification</i> Command Payload.....	463
1074	Figure D-114– Prepay Cluster Client Server Example	466
1075	Figure D-115– <i>Select Available Emergency Credit</i>	494
1076	Figure D-116– Change Debt Command Payload.....	495
1077	Figure D-117– Emergency Credit Setup Command Payload	497
1078	Figure D-118– Consumer Top Up Command Payload	498
1079	Figure D-119– Credit Adjustment Command Payload	499
1080	Figure D-120– Change Payment Mode Command Payload	500
1081	Figure D-121– Get Prepay Snapshot Command Payload	501
1082	Figure D-122– Get Top Up Code Log Command Payload.....	502
1083	Figure D-123– Set Low Credit Warning Level Command Payload	503
1084	Figure D-124– <i>GetDebtRepaymentLog</i> Command Payload	503
1085	Figure D-125– Set Maximum Credit Level Command Payload.....	505
1086	Figure D-126– Set Overall Debt cap Command Payload	506
1087	Figure D-127– Publish Prepay Snapshot Command Payload	507
1088	Figure D-128– Debt/Credit Status Command Payload	509
1089	Figure D-129– Change Payment Mode Response Command Payload.....	510
1090	Figure D-130– Consumer Top Up Response Command Payload.....	511
1091	Figure D-131– Publish Top Up Log Command Payload	513
1092	Figure D-132– Top Up Code Payload.....	513
1093	Figure D-133– Publish Debt Log Command Payload	514
1094	Figure D-134– Format of a Debt Payload Record	514
1095	Figure D-135– Prepayment Credit Status Attribute Explained.....	517
1096	Figure D-136–Calendar Cluster	521

1097	Figure D-137– Recommended Calendar Command Sequence	523
1098	Figure D-138– <i>PublishCalendar</i> Command Payload	526
1099	Figure D-139– <i>PublishDayProfile</i> 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– <i>PublishWeekProfile</i> Command Payload.....	532
1104	Figure D-144– <i>PublishSeasons</i> Command Payload.....	533
1105	Figure D-145– Season Entry Sub-Payload.....	534
1106	Figure D-146– <i>PublishSpecialDays</i> Command Payload	535
1107	Figure D-147– <i>SpecialDayEntry</i> Sub-Payload	536
1108	Figure D-148– <i>CancelCalendar</i> Command Payload	536
1109	Figure D-149– <i>GetCalendar</i> Command Payload	538
1110	Figure D-150– <i>GetDayProfiles</i> Command Payload.....	539
1111	Figure D-151– <i>GetWeekProfiles</i> Command Payload	539
1112	Figure D-152– <i>GetSeasons</i> Command Payload	540
1113	Figure D-153– <i>GetSpecialDays</i> Command Payload	541
1114	Figure D-154– Device Management Cluster Client/Server Example.....	544
1115	Figure D-155– <i>RequestNewPassword</i> 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– <i>Publish Change of Tenancy</i> Command Payload	555
1119	Figure D-159– <i>Publish Change of Supplier</i> Command Payload.....	556
1120	Figure D-160– <i>RequestNewPasswordResponse</i> Command Payload.....	558
1121	Figure D-161– <i>Update SiteID</i> Command Payload.....	559
1122	Figure D-162– <i>Set Event Configuration</i> Command Payload	560
1123	Figure D-163– ‘ <i>Apply by List</i> ’ Sub-Payload	561
1124	Figure D-164– ‘ <i>Apply by Event Group</i> ’ Sub-Payload.....	561
1125	Figure D-165– ‘ <i>Apply by Log Type</i> ’ Sub-Payload	561
1126	Figure D-166– ‘ <i>Apply by Configuration Match</i> ’ Sub-Payload	562
1127	Figure D-167– <i>Get Event Configuration</i> Command	562
1128	Figure D-168– <i>Update CIN</i> 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– <i>Get Event Log</i> Command Payload	595
1132	Figure D-172– <i>Clear Event Log Request</i> Command Payload	596
1133	Figure D-173– <i>Publish Event</i> Command Payload	597
1134	Figure D-174– <i>Publish Event Log</i> Command Payload	598
1135	Figure D-175– <i>Publish Event Log</i> Sub-Payload	599
1136	Figure D-176– <i>Clear Event Log Response</i> Command Payload	600
1137	Figure D-177– Energy Management Cluster	602
1138	Figure D-178– <i>Manage Event</i> Command Payload	607

1139	Figure D-179— <i>Report Event Status</i> Command Payload	609
1140	Figure D-180—MDU Pairing Cluster.....	611
1141	Figure D-181—Pairing Response Command Payload.....	612
1142	Figure D-182—Pairing Request Command Payload	614
1143	Figure E-1—Smart Energy Device Class Reference Example.....	618
1144	Figure E-2—Correctly Overlapping Events	619
1145	Figure E-3—Correct Superseding of Events	620
1146	Figure E-4—Superseded Event for a Subset of Device Classes	621
1147	Figure E-5—Ending Randomization Between Events.....	622
1148	Figure E-6—Start Randomization Between Events	623
1149	Figure E-7—Acceptable Gaps with Start and Stop Randomization.....	624
1150		

1151 Participants

1152

1153 The following is a list of those who were members of the ZigBee PRO Technical Sub-
1154 committee (PRO-TSC) leadership when this document was released:

1155 **Ezra Hale:** Chair

1156 **Phil Jamieson:** Vice-Chair

1157

1158 When the document was released, the ZigBee PRO Energy Domain Working Group leadership
1159 was composed of the following members:

1160 **Larry Kohrmann:** Chair

1161 **Ian Winterburn:** Vice-Chair

1162 **Rob Alexander & Ian Winterburn:** Technical Editors

1163 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 Huggins	John Prater	Jack Worsnop
Rick Enns	Jeffrey King	Yuri Shteinman	

1164 **Document History**

1165

1166 Table 1.1 shows the change history for this specification.

1167

1168

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 <i>Load Profile</i> 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 <i>Get Current Price</i> 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.

11		<p>Third pass at comment resolution. Please refer to document #075424r05ZB for changes.</p> <p>Moved SE Cluster definitions into the annex D.</p> <p>Significant changes in the Security related sections.</p> <p>Best practice information added to more sections.</p> <p>Updated Annex E covering overlapping event examples.</p> <p>Corrected issues relating to the following CCBs:</p> <ul style="list-style-type: none"> CC-900 [SE] Randomizing Price Events CC-901 [SE] Message Cluster Start Time CC-902 [SE] New Status Field for <i>Get Profile Response</i> 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 <i>Report Event Status</i> Command CC-906 [SE] Additional Description of Device Class bits 0 and 1 CC-907 [SE] Array vs. Series of Intervals for <i>Get Profile Response</i> Command CC-908 [SE] Randomization of <i>Report Event Status</i> 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 <i>Publish Price</i> 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.

15	<p>Corrected issues related to the following CCBs (from errata document 084914r05):</p> <p>CC-964 ZigBee Cluster Library reference doesn't contain the revision number.</p> <p>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.</p> <p>CC-966 The Identify cluster should be Optional, not mandatory.</p> <p>CC-967 The Common Features and Functions table incorrectly calls out the binding and service discovery requests as mandatory items.</p> <p>CC-968 Future definitions of fields added to the end of commands are to be treated as reserved fields.</p> <p>CC-973 Addition of Greenhouse Gas (CO₂) pricing information to the <i>Publish Price</i> command.</p> <p>CC-974 Addition of Supply Limit tracking in the Metering Cluster.</p> <p>CC-980 Correct and describe CRC Algorithm used for Installation Codes.</p> <p>CC-981 Correct the Installation Codes text examples and provide example source code for testing/using the MMO Hash Algorithm.</p> <p>CC-982 Attributes <i>CurrentPartialProfileIntervalValueDelivered</i> and <i>CurrentPartialProfileIntervalValueReceived</i> do not list default values or mandatory/optional status.</p> <p>CC-983 Attributes <i>Power Factor</i>, <i>ReadingSnapShotTime</i>, <i>CurrentMaxDemandDeliveredTime</i>, and <i>CurrentMaxDemandReceivedTime</i> are incorrectly replicated in another section.</p> <p>Corrected issues related to the following CCBs:</p> <p>CC-984 Addition of Key Establishment test vectors.</p> <p>CC-986 Addition of metering device types to the simple metering cluster attribute <i>MeteringDeviceType Enumeration</i>.</p> <p>CC-993 Initiate Key Establishment Request and Response Payload Format field names need to match field names defined in Payload Format figures.</p>
----	---

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

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

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

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

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

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

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

19	1.2a	<p>Changes resulting from the SE1.2 MRD (11-5543) and TRD (11-5758), together with the 'Blackwall' changes identified in 14-0046. These include:-</p> <ul style="list-style-type: none"> - 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) <p>CCB 1501 - All ZCL(general) clusters except those in Table 5.13 should be definitively prescribed to use a specific security key</p> <p>CCB 1511 - Tier Label Attribute Set</p> <p>CCB 1679 - Ambiguous definition for when block n vs block n+1 is active</p> <p>CCB 1746 - Incorrect Maximum TC Keep-Alive Interval</p> <p>CCB 1886 - IntervalReadReportingPeriod - Remove Attribute from the Spec</p> <p>CCB 1939 - TunnelID reference in RequestTunnel command should be ProtocolID</p> <p>CCB 1999 - Heating and Cooling Mandatory Attributes</p> <p>CCB 2009 - Publish Topup Command</p> <p>CCB 2010 - Handling of Incorrect Issuer Event ID Fields</p>
----	------	---

1169

1 Introduction

1170

1.1 Scope

1171 This Standard defines device descriptions and standard practices for “Smart Energy” applications
1172 needed in a Smart Energy based residential or light commercial environment. Installation
1173 scenarios range from a single home to an entire apartment complex. The key application domains
1174 are metering, pricing (with associated scheduling), messaging and demand response and load
1175 control. Support applications are provided for commissioning and managing such networks.
1176 Provision is made for operating in both Credit and Prepayment modes. Other applications will be
1177 added in future versions.

1178

1.2 Purpose

1179 This specification provides standard interfaces and device definitions to allow interoperability
1180 among ZigBee devices produced by various manufacturers of electrical equipment, meters,
1181 and Smart Energy enabling products.

1182

1.3 Provisional Features

1183 Some of the features in this version of this specification are provisional and non-
1184 certifiable. The text regarding these features may change before reaching certifiable
1185 status. The features consist of the following items:

- 1186 • Metering cluster attribute 0x0012 (*VolumePerReport*)
- 1187 • Configuration of Notification Schemes/Flags within the Metering cluster
- 1188 • The Price cluster client attributes.
- 1189 • Price cluster Critical Peak Pricing (CPP)
- 1190 • Messaging cluster *Cancel All Messages* and *GetMessageCancellation* commands
- 1191 • Tunneling cluster Flow Control option.
- 1192 • Trust Center Swapout behaviors.
- 1193 • Energy Management cluster.
- 1194 • Federated Trust Center, including the MDU Pairing Cluster
- 1195 • Coexistence with devices using other Profiles
- 1196 • ‘Enhanced’ Inter-PAN transmission mechanism

1198 2 References

1199 2.1 References

1200 The following standards and specifications contain provisions, which through reference in
1201 this document constitute provisions of this specification. All the standards and specifications
1202 listed are normative references. At the time of publication, the editions indicated were valid. All
1203 standards and specifications are subject to revision, and parties to agreements based on this
1204 specification are encouraged to investigate the possibility of applying the most recent editions
1205 of the standards and specifications indicated below.
1206

1207 2.1.1 ZigBee Alliance Documents

- 1208
- 1209 [B1] ZigBee document 07-5123-04, ZigBee Cluster Library Specification, ZigBee Cluster
1210 Library Development Board.
- 1211
- 1212 [B2] ZigBee document 064309r04, Commissioning Framework
- 1213
- 1214 [B3] ZigBee Document 05-3474-20, The ZigBee Specification, ZigBee
1215 Technical Steering Committee (TSC)
- 1216
- 1217 [B4] ZigBee Document 03084r00, ZigBee Key Establishment Proposal
1218 Certicom
- 1219
- 1220 [B5] ZigBee 075297r04, Proposal for Inter-PAN Exchange of Data in ZigBee
- 1221
- 1222 [B6] ZigBee document 095343r01, Installation Code Sample Source Code
- 1223
- 1224 [B7] ZigBee document 08006r05, ZigBee 2007 Layer PICS and Stack Profiles,
1225 ZigBee Core Stack Working Group
- 1226
- 1227 [B8] Over the Air Upgrade Cluster Spec. 09-5264-23
- 1228
- 1229 [B9] Over the Air Upgrade Cluster test spec: 09-5473-07
- 1230
- 1231 [B10] Over the Air Upgrade Cluster PICs: 09-5284-09
- 1232

1233 2.1.2 External Reference Documents

- 1234
- 1235 [B11] Institute of Electrical and Electronics Engineers, Inc., IEEE Std. 802.15.4 2003, IEEE
1236 Standard for Information Technology Telecommunications and Information Exchange
1237 between Systems - Local and Metropolitan Area Networks - Specific Requirements Part
1238 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications
1239 for Low Rate Wireless Personal Area Networks (WPANs). New York: IEEE Press. 2003
1240

- 1241 [B12] ANSI X9.62-2005, Public Key Cryptography for the Financial Services Industry:
1242 The Elliptic Curve Digital Signature Algorithm (ECDSA), American Bankers Association.
1243 Available from <http://www.ansi.org>.
- 1244
- 1245 [B13] ANSI X9.63-2001, Public Key Cryptography for the Financial Services Industry -
1246 Key Agreement and Key Transport Using Elliptic Curve Cryptography, American Bankers
1247 Association, November 20, 2001.
1248 Available from <http://www.ansi.org>.
- 1249
- 1250 [B14] NIST Special Publication 800-56A, Recommendation for Pair-Wise Key
1251 Establishment Schemes Using Discrete Logarithm Cryptography (Revised), March 2007.
1252 Available from <http://csrc.nist.gov>.
- 1253
- 1254 [B15] NIST Special Publication 800-38C, Recommendation for Block Cipher Modes of
1255 Operation: The CCM Mode for Authentication and Confidentiality, May 2004.
1256 Available from <http://csrc.nist.gov>.
- 1257
- 1258 [B16] FIPS Pub 197, Advanced Encryption Standard (AES), Federal Information
1259 Processing Standards Publication 197, US Department of Commerce/N.I.S.T, Springfield,
1260 Virginia, November 26, 2001.
1261 Available from <http://csrc.nist.gov>.
- 1262
- 1263 [B17] FIPS Pub 198, The Keyed-Hash Message Authentication Code (HMAC), Federal
1264 Information Processing Standards Publication 198, US Department of Commerce/N.I.S.T.,
1265 Springfield, Virginia, March 6, 2002.
1266 Available from <http://csrc.nist.gov>.
- 1267
- 1268 [B18] Standards for Efficient Cryptography: SEC 1 (working draft) ver 1.7: Elliptic Curve
1269 Cryptography, Certicom Research, November 13, 2006.
1270 Available from <http://www.secg.org>
- 1271
- 1272 [B19] Standards for Efficient Cryptography: SEC 4 (draft) ver 1.0: Elliptic Curve
1273 Cryptography, Certicom Research, January 24, 2013.
1274 Available from <http://www.secg.org>
- 1275
- 1276 [B20] RFC 3280: Internet X.509 Public Key Infrastructure: Certificate and Certificate
1277 Revocation List (CRL) Profile. IETF, April 2002.
1278 Available from <http://www.ietf.org>
- 1279
- 1280 [B21] DUPLICATE OF [B19] - REMOVED
- 1281
- 1282 [B22] RFC 3280: Internet X.509 Public Key Infrastructure: Certificate and Certificate
1283 Revocation List (CRL) Profile. IETF, April 2002. Available from <http://www.ietf.org>
- 1284
- 1285 [B23] Standards for Efficient Cryptography: SEC 2 (Final) ver 2.0: Recommended Elliptic
1286 Curve Domain Parameters

1287

3 Definitions

1288

3.1 Conformance Levels

1289

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.

1290

1291
1292

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

1293
1294
1295

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”).

1296

1297

1298

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”).

1303

1304

3.2 ZigBee Definitions

1305

1306

1307

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

1308

1309

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

1310

1311

1312

1313

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.

1314

1315

1316

1317

1318

1319

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.

1320

1321

Node: Same as a unit.

1322

1323

1324

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.

1325

1326

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

1327

1328

1329

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.

1330
1331 **ZigBee coordinator:** An IEEE 802.15.4-2003 PAN coordinator.
1332
1333 **ZigBee end device:** an IEEE 802.15.4-2003 RFD or FFD participating in a ZigBee network,
1334 which is neither the ZigBee coordinator nor a ZigBee router.
1335
1336 **ZigBee router:** an IEEE 802.15.4-2003 FFD participating in a ZigBee network, which is not
1337 the ZigBee coordinator but may act as an IEEE 802.15.4-2003 coordinator within its
1338 personal operating space, that is capable of routing messages between devices and supporting
1339 associations.

1340

3.3 Smart Energy Definitions

1341 **Delivered:** In the context of metering, the term “Delivered” refers to the quantity of a commodity
1342 that was delivered to the customer from the utility.

1343 **Export:** In the context of metering, the term “Export” has the same meaning as “Received”.

1344 **Friendly Credit Period:** A duration of time where delivery rules for supply interruption
1345 connected with the accounting functions of a meter are temporarily suspended. Also known as
1346 “Non-Disablement Period”.

1347 **Note1** - When a meter enters into a Friendly Credit Period with a useable positive credit
1348 balance, the consumer will be allowed to consume energy for the duration of the Friendly
1349 Credit Period, regardless of their credit status while in that period. If, however, the consumer
1350 had already run out of credit and supply was interrupted before entering into the Friendly
1351 Credit Period, they will not be allowed to reconnect without first adding suitable additional
1352 credit.

1353 **Note2** - At the end of the Friendly Credit Period, the normal delivery rules connected with the
1354 accounting functions of the meter will be resumed, and if the meter’s credit balance has
1355 dropped below the disablement threshold during the Friendly Credit Period, then the meter
1356 will disconnect upon resuming normal delivery rules.

1357 **Import:** In the context of metering, the term “Import” has the same meaning as “Delivered”.

1358 **Received:** In the context of metering, the term “Received” refers to the quantity of a commodity
1359 that was received by the utility from the customer.

1360 **Top-up:** An operation or transaction resulting in the credit balance held on a prepayment meter
1361 to be increased by use of a credit token. **NOTE** - In IEC terminology, the associated term ‘vend’
1362 would normally relate to a transaction in conjunction with a vending system at a point of sale,
1363 resulting in the creation of a token that can be transported by means of a physical or virtual token
1364 carrier.

1365 **Unique Transaction Reference Number (UTRN):** A numeric or alpha-numeric code used as a
1366 virtual token carrier for a payment metering system. These codes are issued by points of sale and
1367 represent an encoded and secured credit value or special action that the meter must act upon
1368 within its own application code. The functionality of payment metering tokens is outside the
1369 scope of the Smart Energy standard, however further information and examples of such tokens
1370 can be seen in IEC 62055-41.

1371
1372

1373

4 Acronyms And Abbreviations

1374

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

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

1378 **5 Profile Description**

1379

1380 **5.1 A ZigBee Smart Energy Network**

1381

1382 The Smart Energy market requires two types of ZigBee networks for metering and energy
1383 management. These include neighborhood area networks for meters, using ZigBee for sub-
1384 metering within a home or apartment, and using ZigBee to communicate to devices within
1385 the home. Different installations and utility preferences will result in different network
1386 topologies and operation and this profile must allow for these differences. However, each of
1387 these networks will operate using the same Basic Principles to ensure interoperability.
1388

1389 Because of the type of data and control within the Smart Energy network, application
1390 security is a key requirement. The application will use link keys which are optional in the ZigBee
1391 and ZigBee Pro stack profiles but are required within a Smart Energy network. The Trust Center
1392 and all devices on the Smart Energy network must support the installation and use of these keys
1393 as described in the security section.
1394

1395 Metering networks are primarily installed by specialized service personnel, but other devices in
1396 the network may be added by home owners, or home automation professionals who may not
1397 have any ZigBee expertise. Installation concepts must be easy and uniform across Smart Energy
1398 device manufacturers.
1399

1400 Smart Energy networks could include both ZigBee 2007 and ZigBee 2007 Pro nodes. It is
1401 recommended the majority of the nodes in the network should be based on one stack profile or
1402 the other to get consistent performance. ZigBee Smart Energy certified products must be
1403 based upon a ZigBee Compliant Platform (ZCP). If the Smart Energy profile resides in
1404 conjunction with a private profile, the product should be ZigBee Manufacturer Specific
1405 Profile (MSP) certified and must be Smart Energy ZCP certified. This additional certification
1406 provides a reassurance that the underlying stack is behaving properly and the application is not
1407 abusive to the network.
1408

1409 Smart Energy networks will not interact with a consumer ZigBee Home Area Network unless a
1410 device is used to perform an “application level bridge” between the two profiles or the HA
1411 devices satisfy the Smart Energy profile security requirements. This is due to the higher security
1412 requirements on the Smart Energy network that are not required on a Home network. However,
1413 it is expected that Home Automation devices that are extended to include the Smart Energy
1414 profile can still operate in a home network.
1415

1416 The ZigBee Smart Energy Network makes possible networks such as the following:
1417

1418
1419
1420
1421
1422
1423
1424

1425 AMI Server
1426
1427
1428
1429
1430
1431
1432

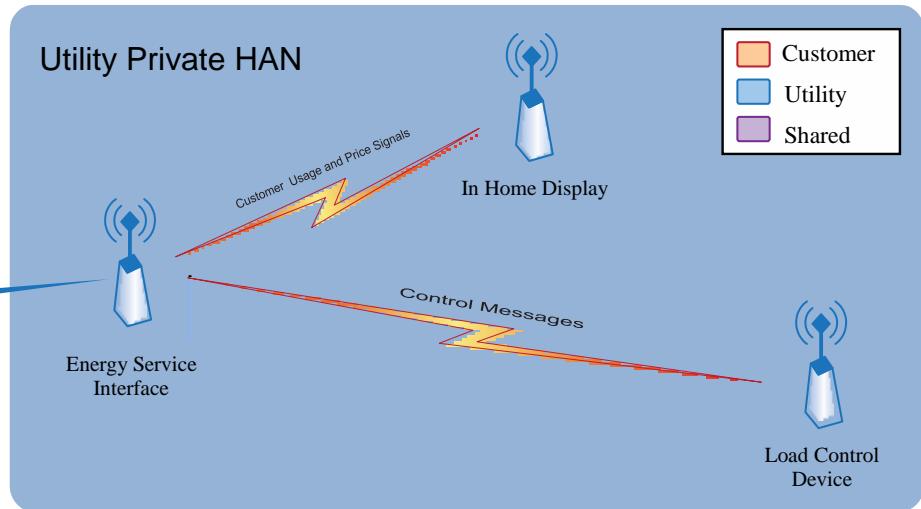


Figure 5-1 Utility Private HAN

1433
1434
1435
1436 Utility Private HAN might include an in-home display, or a load control device working in
1437 conjunction with an energy service interface, but it would not include any customer controlled
1438 devices.
1439

1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451

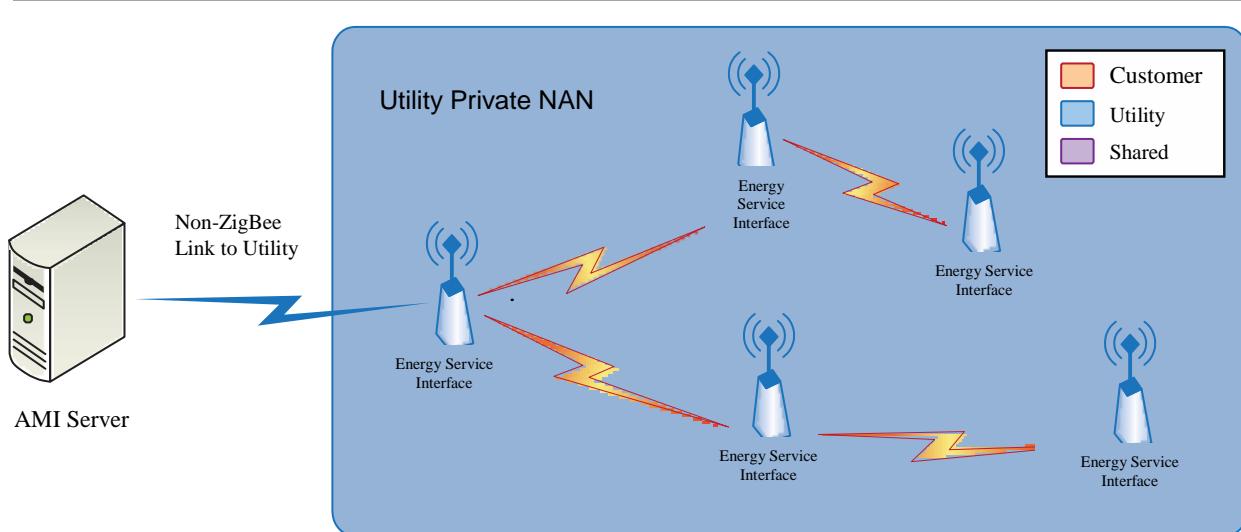


Figure 5-2 Utility Private NAN

1452
1453
1454
1455 Utility Private ZigBee network might also be used as a NAN, where ZigBee provided the
1456 primary communications for a Smart Energy deployment.

1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471

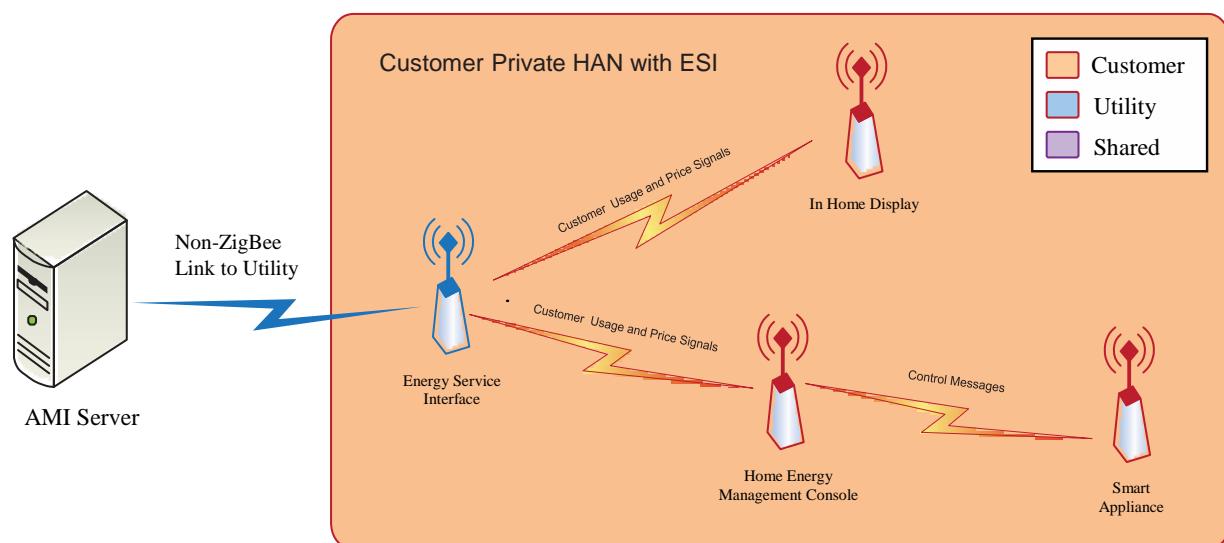


Figure 5-3 Customer Private NAN

1472
1473
1474 ESI provided by utility, but limited to the role of information provider (Usage and Pricing)
1475 into a customer HAN that utilizes an Energy Management Console for conveying or
1476 controlling local devices. An example is controlling a smart appliance based upon a pricing
1477 signal.
1478

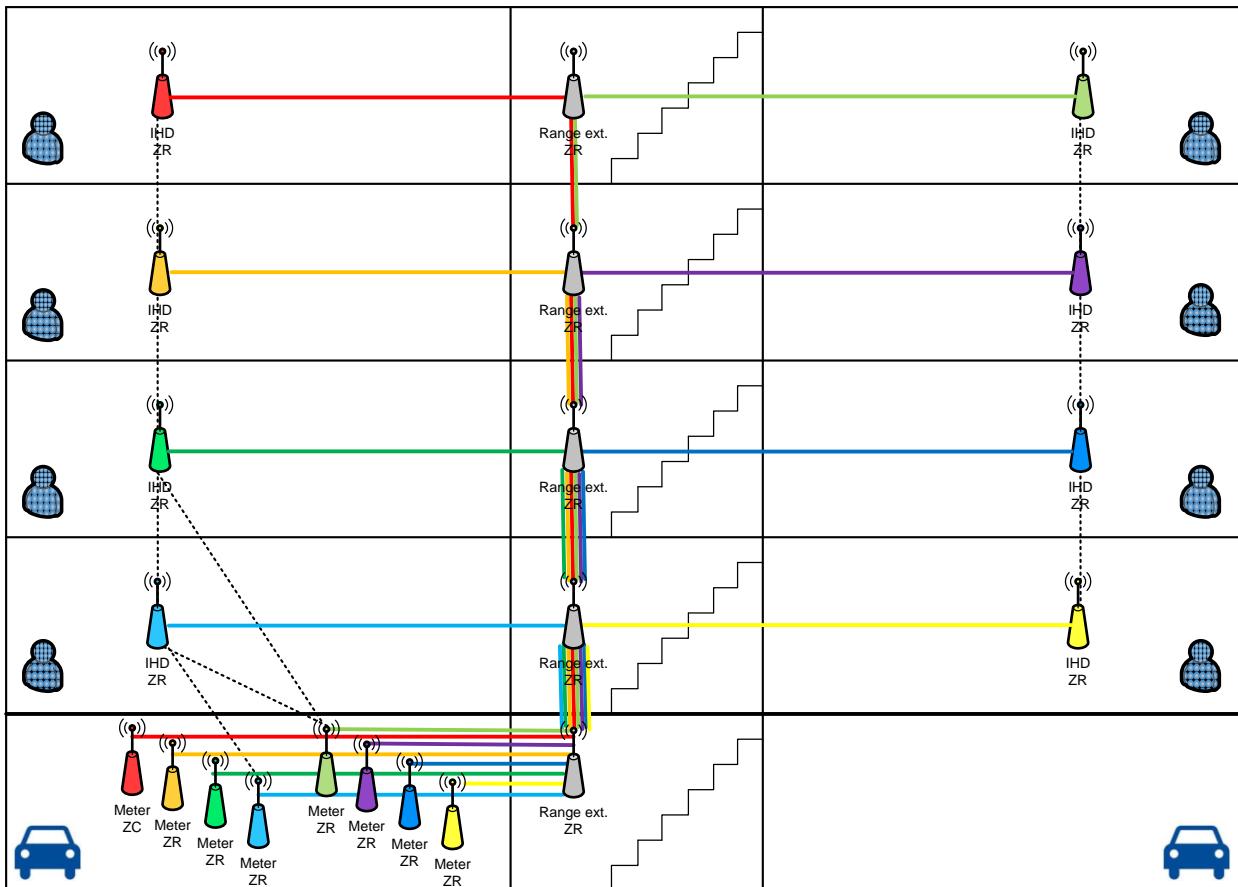


Figure 5-4 An Example of a MDU Setup

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.

- 1495 • Fragmentation is required. Please refer to 5.3.8 regarding fragmentation sizes and parameter
1496 settings.

1497

1498 **5.2.1 MAC Data Polling (NLME_Requests)**

1499 MAC Data polling is required by all sleepy end devices to operate correctly in a ZigBee Pro
1500 network. Smart Energy puts no restrictions on the frequency of MAC data polls. The choice of
1501 how frequently data polling is done will be based on individual product design considerations
1502 to reduce power consumption. However the following are a set of recommendations to ensure
1503 correct operation in the network:

1504

- 1505 • The MAC data polling rate should be dynamic based on the device's operating state. It is
1506 recommended it has at least two rates, a fast rate and a slow rate.
- 1507 • The ZigBee specification only requires that parent devices buffer a single message for 7.5
1508 seconds. This single buffer applies to all sleepy end devices. Therefore a sleepy device
1509 should poll more frequently than once per 7.5 seconds in order to be able to retrieve a
1510 buffered message.
- 1511 • When the device is waiting for an active response message such as an APS
1512 acknowledgement, or a ZCL response, or participating in a multi-message protocol, it should
1513 poll at its fast rate. This fast rate is recommended to be at least once every 3 seconds.
- 1514 • When the device is not actively waiting for messages it can poll at its slow rate. For example
1515 once per hour. This ensures it still has a connection with the network and with its parent.

1516

1517 During initial joining to the Smart Energy network, including key establishment and service
1518 discovery, it should poll at its fast rate.

1519

1520 **5.2.2 Application Level Queries**

1521

1522 It is expected that client devices will periodically send application level queries to servers to
1523 retrieve data. This may be done for example by thermostats querying the current price, or an in-
1524 home display to show the current reading of a meter.

1525

1526 Due to the fact that all ZigBee devices within the HAN utilize a shared medium for sending and
1527 receiving data it is recommended that devices do not saturate the network with frequent queries
1528 for data that does not change often. As a general rule, but not a requirement, it is
1529 recommended that devices do not initiate more than 1 query per second. This
1530 recommendation does not apply to responses generated locally due to the receipt of remote
1531 device requests. In addition, it is possible that the device may need to generate a burst of

1532 traffic and exceed this recommendation. This bursting period should be very limited and
 1533 followed by a period of reduced traffic respecting the above guidelines.
 1534

1535 **5.2.3 ZigBee Coordinator and Trust Center Recommendations**

- 1536
- 1537 • In a Smart Energy based HAN network the Trust Center shall be the Coordinator (short
 1538 address 0x0000).
 - 1539 • In a Smart Energy based HAN network the Trust Center shall be an ESI in the network.
 - 1540 • In a Smart Energy based NAN the backhaul point is likely to be the coordinator and trust
 1541 center.
- 1542

1543 **5.3 Startup Attribute Set (SAS)**

1544 In order to ensure interoperability, all ZigBee Smart Energy devices shall implement
 1545 compatible Startup Attribute Sets (SAS) as defined in this specification. This does not
 1546 mean that the set must be modifiable through a commissioning cluster, but that the device
 1547 must internally implement these stack settings to ensure compatibility and consistent user
 1548 experience. The startup set parameters described by the commissioning cluster in [B2] provide
 1549 a good basis to specify a Smart Energy start up set.

1550

1551 Because Smart Energy Devices are likely to be preconfigured at a warehouse and installed by a
 1552 technician, specific start up set values may be established by a particular utility or service area
 1553 and these startup set values used in place of those below for installation. The startup set values
 1554 that would be expected to be set by the installer are noted below.

1555

1556 **5.3.1 Startup Parameters**

1557 The startup parameters and their default values are listed in Table 5-1.

1558

Table 5-1 – Startup Parameters

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

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 <i>ExtendedPANId</i> 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	0x00000000000000000000 000000 00001 if the Key Establishment Cluster is being used to install a link key Installer provided if using preconfigured link keys	

Network Key	0x00000000000000000000 000000 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

1560

1561 **5.3.2 Join Parameters**

1562 The join parameters and their default values are listed in Table 5-2.

1563

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.

	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

5.3.3 Security Parameters

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.	
TrustCenterNetworkKey	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

1569 **5.3.4 End Device Parameters**

1570 The end device parameters and their default values are listed in Table 5-4.

1571 **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.

1572

1573 **5.3.5 Link Status Parameters**

1574 The link status parameters and their default values are listed in Table 5-5.

1575 **Table 5-5 – Link Status Parameters**

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

1576

1577

1578

1579 **5.3.6 Concentrator Parameters**

1580 The concentrator parameters and their default values are listed in Table 5-6.

1581 **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.

1582

1583 **5.3.7 APS Transport Parameters**

1584 The APS transport parameters and their default values are listed in Table 5-7.

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

5.3.8 APS Fragmentation Parameters

For fragmentation there are application settings from the APS IB that must be defined by the 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

In addition the Maximum Incoming Transfer Size Field in the Node descriptor defines the largest ASDU that can be transferred using fragmentation. For the Smart Energy Profile the default value shall be set to 128 bytes. Maximum ASDU size allowed is specified in [B3] and dictated by solution needs and RAM capacities of the communicating devices.

It is highly recommended all devices first query the Node Descriptor of the device it will communicate with to determine the Maximum Incoming Transfer Size (if ASDU size is greater than 128 bytes). This will establish the largest ASDU that can be supported with fragmentation. The sending device must use a message size during fragmentation that is smaller than this value.

1601

For additional information regarding the use of fragmentation in relation to the Smart Energy Tunneling cluster, please refer to Annex D.6.

1604

5.3.9 Binding Parameters

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

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

5.4.1 Joining with Preinstalled Trust Center Link Keys

1617 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.

- 1637 **6** The device joins the network and is sent the network key encrypted with the key-transport key
1638 derived for the preinstalled trust center link key. The procedure for doing this is detailed in
1639 Annex F, also reference [B3] section 4.5.4 on key-transport keys and [B3] section 4.4.1 on
1640 frame security for the APS layer.
- 1641 **7** After completion of the joining procedure, the device must use the Key Establishment Cluster
1642 to establish a new link key with the trust center. The device shall use its security credentials
1643 when initiating the key establishment protocols.
- 1644 **8** The trust center of the network has the option of later updating the trust center link keys with
1645 devices in the network as desired by the application using the Key Establishment Cluster.
1646 Updating security keys should be an infrequent operation.
- 1647 **9** Once joining is completed, the list of authorized devices in the Trust Center should be updated,
1648 please refer to sub-clause 5.4.1.1, “Best Practices for Tracking Registered Devices”.

1649

1650

1651 **5.4.1.1 Best Practices for Tracking Registered Devices**

1652 In order to properly track Smart Energy Devices and communicate device registration
1653 status to upstream systems, Trust Centers (ESIs) should maintain a list of authorized devices. It
1654 is also recommended that Trust Centers maintain the following items for each of the registered
1655 devices:

1656

1657

1658 **1 Client EUI64**

1659 **2 Client Installation Code**

1660 **3 Registration Status**

1661 **4 Time and Date Stamps**

1662 **5 Supported Cryptographic Suites**

1663 With the exception of the cryptographic suite, this information is not exposed through the
1664 ZigBee network. However, the device information is expected to be used to track and
1665 understand ZigBee network connectivity.

1666

1667 **5.4.1.2 Best Practice for Coordinator Permit Joining Broadcasts**

1668 It will be left to the coordinator / administrators of the network to determine when a network
1669 should be allowing joining. However when the network is allowing joining:

- 1670
- 1671 1 At the start of the joining period the coordinator will allow joining and broadcast a permit join
1672 message for the lesser of the permit join period or 254 seconds.
- 1673 2 Every 240 seconds or whenever a device announce is received the coordinator will broadcast a
1674 permit join message for the lesser of the remaining permit join period or 254 seconds.
1675 Administrators of a network shall try to keep the amount of time devices on their networks
1676 allow joining to a minimum.

1677 *Note: sending out a permit join message with a time of 255 (forever) is disallowed due to the risk*
1678 *of not being able to reliably tell devices to stop permitting joining in the future.*
1679

1680 5.4.2 Re-Joining a Secured Network

1681

1682 5.4.2.1 Rejoining Node Operation

1683 When a device is re-joining a secured network, the following steps are used:

1684 1 Permit joining is not required to be on in the network.

1686 2 The device shall attempt a rejoin using the procedure detailed in [B3] Section 3.6.1.4.2 with
1687 network security. The network key and sequence number used will be the ones previously
1688 obtained from the trust center.

1689 3 If the secured rejoin is successful, nothing more is required from the device.

1690 4 If the secured rejoin fails, the device shall attempt a rejoin using the procedure detailed in [B3]
1691 Section 3.6.1.4.2 without network security. The re-joining device is assumed to have
1692 previously joined the network and obtained a link key using the key establishment cluster
1693 procedures. If the device does not have a link key obtained via the key establishment cluster, it
1694 cannot rejoin the network.

1695 5 If the rejoin fails the device may attempt it again. If the device is told to leave the network it
1696 may employ the Joining using the Key Establishment Cluster procedure.
1697

1698 5.4.2.2 Trust Center Operation

1699 When the trust center receives notification that a device has rejoined the network, the following
1700 steps are used:

1701 1 If the device performed a secured rejoin the trust center is not required to take any action.
1702

- 1703 **2** If the device performed a rejoin the trust center shall determine if the device is authorized to be
1704 on the network. The trust center should send out an updated copy of the network key encrypted
1705 with the corresponding link key.
- 1706 **3** If the trust center determines that the device is not authorized to be on the network, it shall
1707 send an APS *Remove Device* command to the parent of the rejoicing device, with the target
1708 address of the rejoicing device's IEEE address. The parent will then remove that device from
1709 its child table.

1710
1711 *Note: The Trust Center and Router behaviors described in sections beginning at sub-clause
1712 5.4.2.2.1 up until sub-clause 5.4.3 in this revision of this specification are provisional and not
1713 certifiable. This text may change before reaching certifiable status in a future revision of this
1714 specification.*
1715

1716 **5.4.2.2.1 Initiating Re-Registration**

1717 To initiate the re-registration process for a device, the Trust Center (ESI) would invalidate the
1718 Link keys for that device and subsequently cause a re-authentication / authorization to
1719 re-establish Link Keys. The processes required for this activity are:

- 1720 **1** The Trust Center invalidates the Link key by using the APSME-SET primitive.
- 1721 **2** When the Client device detects communication errors due via APS error results or by
1722 experiencing multiple re-try failures, both caused by the invalid Link Keys, it starts the
1723 processes to validate the following conditions:
- 1724 a The Device validates its still part of the network.
- 1725 b Route discovery processes validate communications paths are still in place.
- 1726 **3** If both conditions are true, the Client device attempts a secure re-join outlined in Re-joining a
1727 Secured Network and subsequently refreshes the Link Keys.
- 1728 **4** Re-binding of services take place (if needed).
- 1729 **5** Once Registration is completed, the list of authorized devices in the Trust Center should be
1730 updated, please refer to sub-clause 5.4.1.1.
1731

1732 **5.4.2.2.2 Initiating De-Registration**

1734 To initiate the de-registration process for a device, which is the process of removing a
1735 previously registered device, the Trust Center (ESI) would use the following processes for this
1736 activity:

- 1737 1 The Trust Center (ESI) invalidates the Link key by using the APSME-SET primitive.
- 1738 2 The Trust Center (ESI) informs the Client device to leave the network by calling the NLME-
1739 LEAVE.request primitive.
- 1740 3 The Trust Center (ESI) informs any Routers to remove the Client device by calling the
1741 APSME-REMOVEDevice.request
- 1742 4 The ESI would unbind any services associated with the Client device by calling the APSME-
1743 UNBIND primitive.
- 1744 5 Once de-registration is completed, the list of authorized devices in the Trust Center should be
1745 updated.

1746

1747 **5.4.2.2.3 Trust Center Swap-Out**

1748 *Note: The Trust Center Swap-Out feature in this revision of this specification is provisional
1749 and is not certifiable. This feature set may change before reaching certifiable status in a future
1750 revision of this specification.*

1751 This section describes the requirements for swapping out a Trust Center in a Smart Energy
1752 network. In the Smart Energy Standard, an ESI should act as the coordinator and trust center of
1753 the network. In most deployments the ESI is the meter and therefore the TC. There can only
1754 be one TC in a SE network, although multiple ESIs may exist on the network. The TC (ESI) in
1755 a SE network is responsible for performing authentication and authorization. SE devices which
1756 are allowed to join the network are provisioned on the TC (ESI) from the head-end over the
1757 utility's backhaul connection.

1758 When a TC is replaced the new device is given the extended PAN ID of the previous
1759 network, and the addresses and associated trust center link keys of all the devices from the
1760 previous network. Both the existing devices and the TC treat these keys like installation codes
1761 (unauthorized), which have limited privileges in the network. Once the devices successfully
1762 connect to the new trust center they must re-establish new TC link keys using CBKE.

1763 Existing Smart Energy devices must be upgraded to include behavior that allows them to
1764 detect a failure to communicate with the existing trust center. When it detects this condition a
1765 device will go off in search of another network with the same extended PAN ID as the current
1766 one. If a network is found then the device will perform a first time join using the NWK rejoin
1767 and its current TC link key as the pre-configured key. If the device is able to successfully join

1768 the network then it will immediately initiate CBKE to derive a new link key with the
1769 replacement TC. If that succeeds the device will identify the device's IEEE as the identity of the
1770 new trust center, and begin operating in the new network; the device shall locate any services
1771 that it may have been using.

1772 If it is unable to join to the new network or unable to successfully negotiate CBKE, then the
1773 device will return to its previous network and continue operating. If the trust center is still
1774 unreachable at a later point in time it can perform the above steps again to attempt to find a
1775 new network.

1776 Trust Center Swap-Out is an optional feature and is not required for ZigBee Smart Energy
1777 Certification.

1778

1779 5.4.2.2.3.1 SE Router Requirements

1780 All routers in the network shall be able to identify when the trust center is no longer
1781 accessible in the network. This will be done by periodically sending an APS datagram to the
1782 Trust Center and receiving the APS acknowledgment. The APS datagram shall require
1783 encryption and acknowledgement.

1784 After an extended period where multiple attempts have been made to contact the trust center and
1785 failed to get a response, a device would temporarily drop off the network to go in search of a
1786 network where the trust center was present. The new network may have different network
1787 parameters than the old one, but the extended PAN ID value would always be the same. Those
1788 networks that match all of the parameters of the old network will be filtered out in preference of
1789 a new network with one or more different parameters. This enables the device to find a newer
1790 instance of the existing network.

1791 Once a new instance of the existing network has been found, the device would perform
1792 various procedures to attempt to join that network and authenticate with the new trust center
1793 device. If at any point during the attempt a failure occurred, the device may continue scanning
1794 for networks to join or return to its existing network and continue operating as it had before.

1795 As a last resort, all devices must have a means to return to factory defaults so that they can be
1796 recommissioned. This would involve reverting back to the use of an installation code and
1797 forgetting all previous network and application parameters. This provides a means to reconnect
1798 the device to the existing network when other methods have failed, or decommission the device
1799 and join it to a new network.

1800

1801 5.4.2.2.3.2 Per SE Network Storage Requirements

1802 TC swap-out requires the backup of data to an off-chip device. The data and storage
 1803 requirements are listed in Table 5.10. Backup of the Extended PAN ID should be performed
 1804 once the ESI has been commissioned or the network is formed. Backup of the TC Link Key
 1805 Hash (see sub-clause 5.4.2.2.3.6) should be performed on successful completion of CBKE with
 1806 the TC. TC Link Key updates from subsequent CBKE shall also be backed up. The Install
 1807 Code derived TC Link Key may be backed up when the device is provisioned on to the SE
 1808 network.

1809 **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

1810
 1811
 1812
 1813 **5.4.2.2.3.3 Utility Requirements**

1814 It is expected that the utility is able to store backup data about each Smart Energy network in
 1815 order to facilitate the TC swap-out feature. It is recommended that the list of IEEE (EUI64)
 1816 addresses of devices registered in the network, and their associated installation code, always be
 1817 backed up. This will help to deal with an unexpected situation due to the customer or the utility,
 1818 which requires one or more HAN devices to be recommissioned.

1819
 1820 **5.4.2.2.3.4 Keep Alive Method**

1821 In order to detect the TC is no longer available all SE routers shall implement a keep-alive
 1822 mechanism with the TC. The Key Establishment cluster is mandatory on all SE devices. The SE
 1823 routers shall send an APS encrypted ZCL message on a periodic interval of up to a maximum of
 1824 20 minutes. The minimum polling rate should not be less than 5 minutes. Failure to receive an
 1825 encrypted APS data frame (such as a read attribute response) shall indicate the TC is no longer
 1826 available. If the device fails to receive 3 APS encrypted data frames in a row it shall consider
 1827 the TC no longer accessible and initiate a search for it. Failure of the encryption or frame
 1828 counter shall constitute a failure of the keep-alive.

1829
 1830 **5.4.2.2.3.5 Trust Center Swap-out Process**

1831 The following steps describe the Trust Center swap-out process.

1832 Preconditions: ESI installed and PAN formed.

- 1833 **1** Back up Extended PAN ID to off-chip device (mandatory).
- 1834 **2** SE device provisioned on the ESI and installed.
- 1835 **3** Back up EUI64 and Install Code to off-chip device (optional).
- 1836 **4** SE device performs CBKE successfully, derived TC link key and EUI64 backed up to off-chip
1837 device (mandatory).
- 1838 **5** Any updates to the TC Link Key must be backed up to off-chip device (mandatory).
- 1839 **6** SE device sends periodic APS encrypted command to the TC.
- 1840 **7** ESI replaced with the Extended PAN ID, list of EUI64s and hashed TC link keys restored
1841 from backed up data. (Permit joining is not required to be on in the network). The TC link keys
1842 shall be treated as install code-derived link keys and unauthorized.
- 1843 **8** New TC forms a new network using new network key, new short PAN ID, and backup of
1844 extended PAN ID.
- 1845 **9** SE device detects TC no longer available (see sub-clause 5.4.2.2.3.4).
- 1846 a There are 4 possible cases at this point.
- 1847 i The TC is temporarily unavailable.
- 1848 ii The device missed a network key update.
- 1849 iii The TC changed channels to avoid congestion.
- 1850 iv The TC has been swapped out.
- 1851
- 1852 **10** SE device scans for the current Extended PAN ID (the short PAN ID will probably have
1853 changed).
- 1854 **11** Prior to performing a rejoin the device must backup in local storage its current TC link key,
1855 state of the link key (authorized or unauthorized), network key, and associated NWK and APS
1856 frame counters.
- 1857 **12** If the TC sends a Transport Key message encrypted using the device's existing TC link key and
1858 the device is able to successfully decrypt and rejoin the network, no further operations are
1859 necessary. The device can resume all normal operations.

1860 **13** If the TC sends a Transport Key message encrypted using a 128-bit AES-MMO hash of the TC
1861 link key, then the device shall ignore the frame counter check and accept the new network key.
1862 It will also record the source IEEE address of the sending device as the new TC address. It
1863 shall mark the hashed TC link key as not authorized and treat the link key as an installation
1864 code. It must now perform Key Establishment to fully authenticate itself in the network.

1865 **14** If the key establishment is NOT successful the device may try again immediately. Otherwise it
1866 shall leave that network. It can continue scanning for additional networks to rejoin, or restore
1867 the values of its previous network and resume normal operation.

1868 15 If the Key Establishment is successful then the device can resume normal operation. It may
1869 discard the backup of security data from the old network. The TC must backup hash of the new
1870 TC Link Key for this device to an off-chip device.

1871 **16** If after attempting rejoin with all discovered PANs fails then the device shall fall back to the
1872 existing PAN.

1873 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.

1879 The hashed key shall be created by performing a 128-bit AES-MMO hash on the
1880 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	C0C1C2C3C4C5C6C7C8C9CACBCCCDCCECF
Hashed TC Link Key	A7977E88BC0B61E8210827109A228F2D

1884 542237 Trust Center (ESI/Meter)

1885 Dependencies

1886 The ESI shall support backup and restore of data (including TC link keys) to an off-chip
1887 device.

1888 Routers shall detect the TC is no longer available by sending an APS encrypted command and
 1889 receiving the APS acknowledgement with a maximum periodic interval of 30¹ minutes.

1890 **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

1891

1892 **5.4.3 Devices Leaving the Network**

1893 Upon receipt of an APS update device command indicating a device has left the network the
 1894 trust center shall not remove the trust center link key assigned to that device. This is to prevent
 1895 a device on the network performing a denial of service attack by spoofing the MAC address of
 1896 another node and issuing a false ZigBee *Network Leave* command. Devices should be removed
 1897 from Trust Center authorization and trust center link key lists via out of band methods, i.e. secure
 1898 meter back haul or secure IP interface.

1899 Devices should follow the guidelines for stale keys described in 5.4.5.

1900 **5.4.4 Updating the Network Key**

1901 Periodically the trust center shall update the network key. This allows the trust center to phase
 1902 out a previous instance of the network key so that devices that are no longer on the network will
 1903 not be able to perform a secure rejoin. Those devices must then perform a rejoin, which
 1904 allows the trust center to authorize whether or not they are allowed to be on the network.

1905 When the trust center wishes to update the network key it will broadcast the network key to
 1906 all devices in the network. All devices receiving the key update will store but will not start
 1907 using the new key.

1908 It is assumed that routers will receive the network key update sent by the Trust Center. Sleepy
 1909 end devices are unlikely to get the network key update sent by the Trust Center unless the device
 1910 polls frequently.

1911 After sending an updated network key, the trust center shall wait a minimum of
 1912 nwkNetworkBroadcastDeliveryTime before sending the switch key message. Devices that miss
 1913 the key switch broadcast message will implicitly switch when they receive any network
 1914 message that is encrypted using the new key sequence number.

¹ CCB 1746

1915 Once the network has started using the new key, any device that has missed the key update
1916 message will not be able to communicate on the network. Those devices that missed the key
1917 update must follow the Re-joining a Secured Network procedure.

1918 **5.4.5 Updating the Link Key**

1919 Periodically the trust center may update the link key associated with a particular device. This
1920 allows the trust center to phase out the existing key and refresh it with a new key. The trust
1921 center can decide on its own what the policy is for how long a link key may be used and how
1922 often it should be updated.

1923 Trust Center link keys are used for sending application messages as well as stack commands.
1924 Therefore a trust center cannot simply delete a link key that it wants to update. The trust center
1925 must accept and or send encrypted APS commands to or from a device even if it has retired that
1926 link key from encryption of application data messages. This is especially necessary for sleeping
1927 end devices, which may not have the current network key and need to use their link key to
1928 obtain an updated copy during a rejoin.

1929 When the trust center deems that a particular link key should no longer be used, it shall mark the
1930 key as stale. A stale key shall not be used to send data messages. Devices that receive a
1931 message using a stale key should discard the message and shall not send an APS
1932 acknowledgement to the sender.

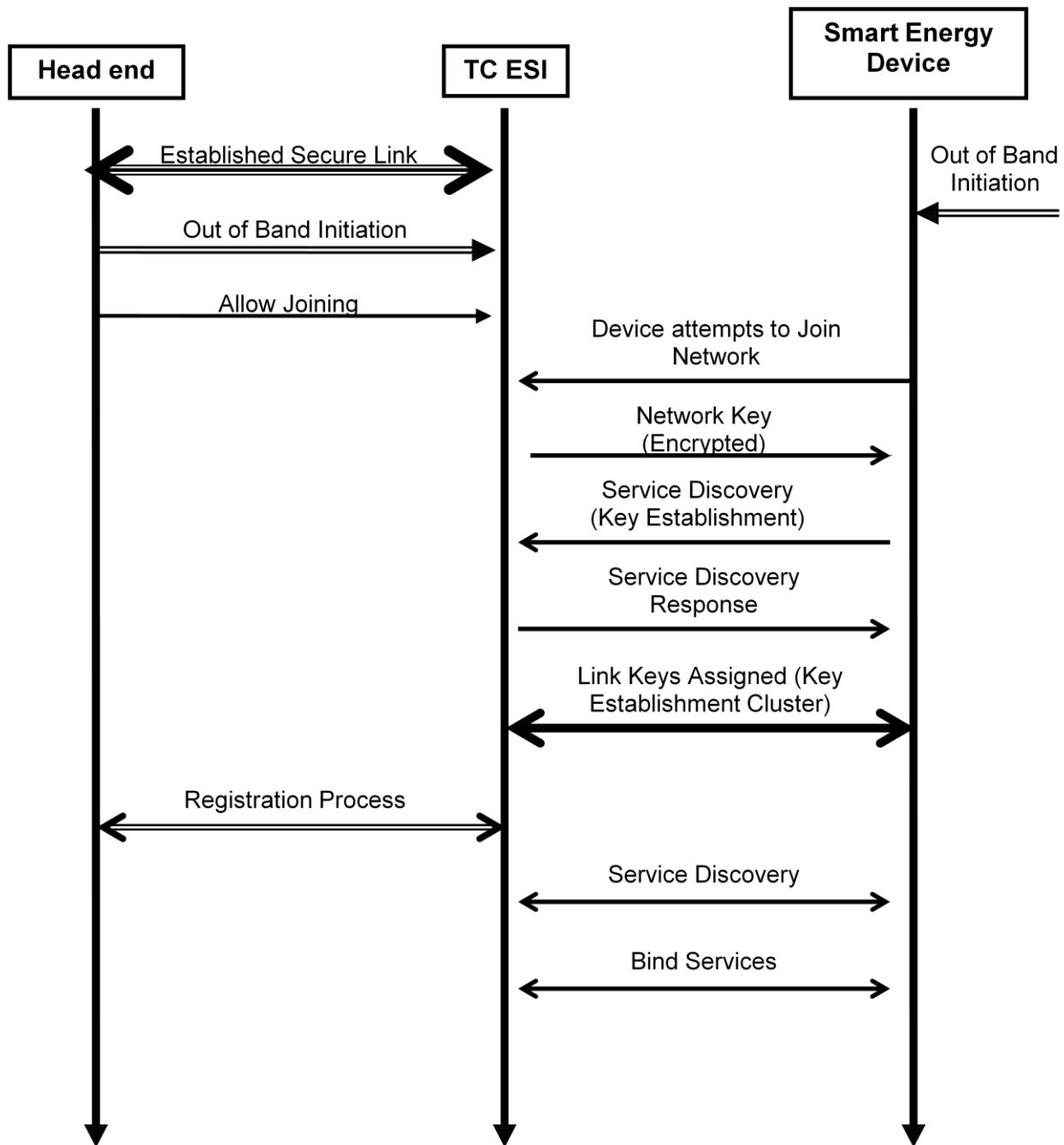
1933 Devices shall accept and process APS commands that are encrypted with a stale key.

1934 When the trust center receives a message encrypted with a stale link key, it shall initiate the key
1935 establishment procedure to negotiate a new link key. Upon successful establishment of the new
1936 link key with the device, the device shall clear the stale indicator for that key.

1937 Devices that are not acting as the trust center may utilize their own policy for retiring and
1938 updating application link keys with other devices that are not the trust center. Those devices are
1939 not required to keep around retired keys and therefore may delete them prior to establishing
1940 an updated link key using the Key Establishment cluster.

1941 **5.4.5.1 Network Joining and Registration Diagram**

1942 Figure 5-5 depicts an example of a successful network startup and certificate exchange (with
1943 pre-established link keys). Please refer to Annex C for further discussions on communication
1944 exchanges and key support.

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

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

-
- 1951 Key Establishment Cluster service discovery process whenever it detects the Smart Energy End
1952 Device fails to do so.
- 1953 After Joining and after Key Establishment:
- 1954 • A device supporting MDUs shall request MDU Pairing Information from the Trust Center
1955 (ESI). If MDU Pairing Information is returned, the device should limit service discovery to
1956 those devices identified in the returned information. If the device has joined a ‘normal’
1957 Smart Energy HAN, no Pairing Information will be returned and the device should perform
1958 service discovery with all devices on the network. Refer to 5.6 and Annex D.13 for further
1959 details.
- 1960 • Client SHALL perform service discovery.
- 1961 • Sleepy devices SHALL perform “get” requests for data they wish to receive and SHOULD
1962 NOT expect to receive unsolicited messages.
- 1963 • If a Client wishes to receive unsolicited messages, Client SHALL follow with attempt(s) to
1964 ZDO Bind Request. A Client does not have to support a binding table.
- 1965 • If Server does not support binding, Server SHALL perform service discovery and register
1966 those devices for unsolicited messages (whether or not they want the messages).
- 1967 • For backward compatibility, Server SHOULD perform service discovery and register those
1968 devices for unsolicited messages (whether or not they want the messages).

1969

1970 **5.4.6 Cluster Usage of Security Keys**

1971 The SE Profile utilizes a higher level of security on the network but not all clusters need to utilize
1972 Application Link keys. All clusters are required to use network layer encryption using the
1973 network key. Table 5-13 identifies the security keys utilized by each cluster:

1974

1975 **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

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

1976

1977

1978 Once a Registered SE device has an Application Link Key established with the ESI, it may also
 1979 establish Application Link Keys with any other device on the SE Network. This is accomplished by
 1980 using the ZigBee service and device discovery process (employing the Network Key). Note that,
 1981 in an MDU network, the service and device discovery process can only commence once MDU
 1982 Pairing information has been acquired, and will then be limited to devices within the same
 1983 ‘virtual HAN’. Regardless of the communication paths, all SE applications shall use and
 1984 validate the Security key usage as listed in Table 5-13. If link key encryption is NOT used but
 1985 required, the receiving device shall generate a ZCL Default Response, employing the Network
 1986 Key, with a FAILURE (0x01) status code.

1987 It is permissible for a device to initiate a ZCL exchange using an application link key even when not
 1988 required. If a device receives a message with link key security even though it is not required as per
 1989 Table 5-13, it shall accept the message. Additionally, if a response is sent then it shall use link key
 1990 encryption.

1991 Unless stated otherwise, any ZCL clusters added to a Smart Energy endpoint shall be APS
1992 encrypted. ZCL clusters without APS encryption shall be located on another (non-Smart Energy)
1993 endpoint. The same restrictions shall also apply to manufacturer-specific clusters.²

1994 **5.4.7 Key Establishment Related Security Policies**

1995 The following are the policies relating to Key Establishment that are recommended for Smart
1996 Energy networks.

1997 **5.4.7.1 Joining**

1999 If the device does not need to perform discovery queries or other non-secure operations after
2000 it joins an SE network and receives the Network Key, it should immediately initiate Key
2001 Establishment with the Trust Center to obtain a new Trust Center Link Key.

2002 If Key Establishment fails with a result of UNKNOWN_ISSUER the device shall leave the
2003 network. A device that does not initiate Key Establishment with the Trust Center within a
2004 reasonable period of time MAY be told to leave depending on the network operator's policy. A
2005 maximum period of 20 minutes is recommended.

2006 Upon successful negotiation of a new Trust Center Link Key the device may communicate using
2007 clusters that require APS security.

2008 **5.4.7.2 Trust Center**

2010 The Trust Center shall keep track of whether a particular device has negotiated a CBKE Trust
2011 Center Link Key, or whether only a preconfigured Trust Center Link Key exists. The Trust
2012 Center shall not use the preconfigured link key to send encrypted APS Data messages to the
2013 device. The Trust Center shall discard any APS encrypted APS Data messages that use the
2014 preconfigured link key, and it shall not send APS Acknowledgments for those messages.

2015 The Trust Center shall accept and send APS Data messages that do not use APS Encryption to a
2016 device that has not negotiated a CBKE Trust Center Link key provided that the security usage
2017 for that cluster allows using only Network layer security (encrypted with the Network Key). See
2018 sub-clause 5.4.6, "Cluster Usage of Security Keys".

2019 The Trust Center is required to be a Smart Energy device. It is required to support Key
2020 Establishment server on at least one endpoint, though it may support it on more than one
2021 endpoint. These endpoints shall be considered to all refer to the same logical ZigBee device
2022 type, in other words the Trust Center. Any negotiation or establishment of a link key on one
2023 endpoint applies globally to the Trust Center as a device and is not specific to an endpoint.

² CCB 1501

2024 The Trust Center shall have a means of adding and removing keys of specific devices that are
2025 part of the Smart Energy network. The specific means of doing this is outside the scope of this
2026 document.

2027 **5.4.7.3 During Joining**

2028 Normal operation of a device in a Smart Energy network requires use of a preconfigured
2029 link key, established by using the Installation Code (refer to sub-clause 5.4.6), to join a ZigBee
2030 Pro network. After joining the network a device is required to initiate key establishment using
2031 ECMQV key agreement with the Trust Center, to obtain a new link key authorized for use in
2032 application messages.

2033 Prior to updating the preconfigured link key using key establishment, the Trust Center shall
2034 not allow Smart Energy messages that require APS encryption. Although the node has a link
2035 key, that node has not been authenticated and thus the key's use is not authorized for application
2036 messages. Its use is still required for certain stack messages (e.g., the APS Command Update
2037 Device) and must be accepted by the trust center.

2038 In order to perform key establishment the device must discover an endpoint on the Trust Center
2039 that supports the Key Establishment server cluster. The joining device shall perform a ZDO
2040 Match Descriptor Request to determine what endpoint to use. This request shall be unicast to
2041 the Trust Center's short address of 0x0000.

2042 When a reply is received, it may contain multiple endpoints that indicate support for the Key
2043 Establishment server. The joining device may use any endpoint to perform key establishment.
2044 Link keys established using key establishment are global to the Trust Center device and are not
2045 specific to a particular endpoint.

2046 Once a node has been authenticated by the Trust Center and obtained an authorized link
2047 key using key establishment, it may communicate with the Trust Center using APS layer
2048 security. The Trust Center should accept valid APS encrypted message using that new link
2049 key. At this point the joining device can communicate to the Trust Center as a Smart Energy
2050 device.

2051 **5.4.7.4 After Joining**

2052 After a node has joined, been authenticated using key establishment, and obtained an authorized
2053 link key, it may need to communicate with other nodes on the network using APS layer
2054 encryption.

2055 Rather than use key establishment with each node on the network, it would be advantageous to
2056 leverage the Trust Center to broker trust with other devices on the network. If two nodes have
2057 both obtained link keys with the Trust Center using key establishment, then they both trust the
2058 Trust Center. Both nodes will use the Trust Center to request a link key with each other. The

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.

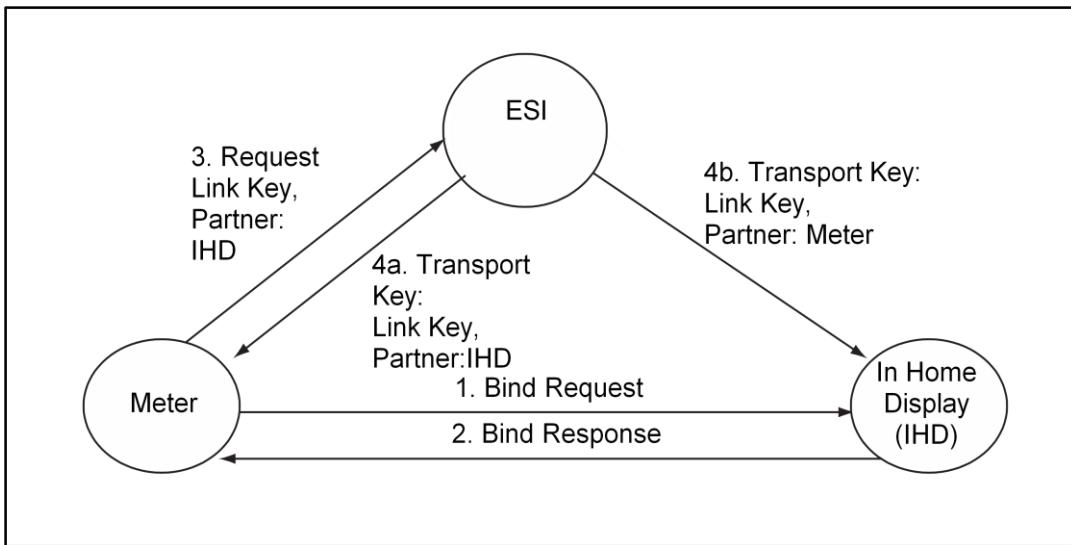


Figure 5-6 Node Communication with Other Nodes on the Network Using APS Layer Encryption

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

2084 terminate the key establishment immediately with a status of NO_RESOURCES. This ensures
 2085 that the ESI authenticates all devices with key establishment after joining, and limits the
 2086 use of key establishment in the network.

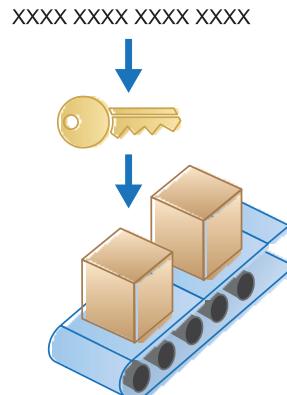
2087 Other ESI devices on the network that are not the trust center would have to go through the same
 2088 procedure as above, contacting the ESI trust center, in order to send/receive messages that require
 2089 APS layer encryption with another node.

2090 **5.4.8 Security Best Practices**

2091
 2092 **5.4.8.1 Out of Band Pre-Configured Link Key Process**
 2093 This section describes the out of band process for establishing pre-configured Trust Center
 2094 link keys, the format of the Installation Code required, and the hashing function used to
 2095 derive the pre-configured link key from the Installation Code.

2096 As portrayed in Figure 5-7, during the manufacturing process a random Installation Code is
 2097 created for each of the Smart Energy devices. This Installation Code is provided for the device in a
 2098 manufacturer-specific way (labeling, etc.) and referenced to during installation. The space of
 2099 installation codes should possess the same randomness properties as a key space. Knowing a
 2100 set of installation codes should not yield any knowledge of another installation code; and
 2101 each installation code should be equally probable. The associated Pre-configured Link Key is
 2102 derived using the hashing function described below and programmed in the device.
 2103
 2104

- 2105 Step 1: An Installation Code is created
 2106 and made available
- 2107 Step 2: The Pre-configured Link Key is
 2108 derived from the Installation
 2109 Code using the Matyas-Meyer-
 2110 Oseas hash function
- 2111 Step 3: The Pre-configured Link Key is
 2112 configured in the device



2120 **Figure 5-7 Smart Energy Device Installation Code Process**

2121
 2122
 2123 As portrayed in Figure 5-8, during the installation process the initial Trust Center Link Key is
 2124 derived from the Installation Code and sent via an out of band communication channel to the

2125 Trust center (ESI). The Trust center uses this Key as the Trust Center Link Key to subsequently
2126 configure the Network Key of the associating device.

2127

2128

2129 Step 1: The Installation Code is sent out of band

2130
2131 Step 2: The Pre-configured Link Key is derived
2132 from the Installation Code using the
2133 Matyas-Meyer-Oseas hash function

2134
2135 Step 3: The Pre-configured Link Key is sent to
2136 the Trust Center using the AMI network

2137

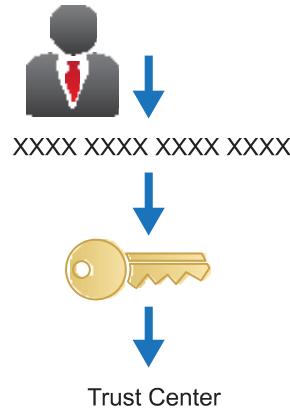
2138

2139

2140

2141

2142



2143 **Figure 5-8 Installation Code Use with the Trust Center**

2144

2145 5.4.8.1.1 Installation Code Format

2146 The Installation Code consists of a 48, 64, 96, or 128 bit number and a 16 bit CRC (using CCITT
2147 CRC standard polynomial $X^{16} + X^{12} + X^5 + 1$). When printed or displayed, Installation Codes
2148 are represented as multiple groups of 4 hexadecimal digits.

2149 48 Bit example:

2150 Installation Code of “83FE D340 7A93 2B70”

2151 Where values 0x83, 0x FE, 0xD3, 0x40, 0x 7A, and 0x93 are used to calculate the CRC16
2152 with the result returning 0x702B.

2153
2154 *Note: The Octet order of the CRC code in the printed Installation code is Least Significant
2155 Octet followed by Most Significant Octet, giving the printed result of “2B70”.*

2156 64 Bit example:

2157 Installation Code of “83FE D340 7A93 9738 C552”

2158 Where values 0x83, 0x FE, 0xD3, 0x40, 0x 7A, 0x93, 0x 97, and 0x38 are used to calculate the
2159 CRC16 with the result returning 0x52C5.

2160

2161 96 Bit example:

2162 Installation Code of “83FE D340 7A93 9723 A5C6 39FF 4C12”

2163 Where values 0x83, 0x FE, 0xD3, 0x40, 0x 7A, 0x93, 0x 97, 0x23, 0xA5, 0xC6, 0x39 and 0xFF
2164 are used to calculate the CRC16 with the result returning 0x124C.

2165

2166

2167
2168
2169 128 Bit example:
2170 Installation Code of “83FE D340 7A93 9723 A5C6 39B2 6916 D505 C3B5”
2171 Where values 0x83, 0xFE, 0xD3, 0x40, 0x7A, 0x93, 0x97, 0x23, 0xA5, 0xC6, 0x39,
2172 0xB2, 0x69, 0x16, 0xD5, and 0x05 are used to calculate the CRC16 with the result
2173 returning 0xB5C3.

2174
2175 5.4.8.1.1.1 CRC Algorithm Information

2176 As stated earlier, the Installation Code CRC calculation is based upon the CRC
2177 16-CCITT algorithm and uses the following parameters:

2178 Length: 16
2179 Polynomial: $x^{16} + x^{12} + x^5 + 1$ (0x1021)
2180 Initialization method: Direct
2181 Initialization value: 0xFFFF
2182 Final XOR value: 0xFFFF
2183 Reflected In: True
2184 Reflected Out: True

2185 Open source implementations of the CRC 16-CCITT algorithm are available on the internet at
2186 sites like SourceForge and others. The source code is also available for download from the
2187 ZigBee document management system [B6].
2188

2189 5.4.8.1.2 Hashing Function

2190 An AES-128 key is derived from the Installation Code using the Matyas-Meyer- Oseas (MMO)
2191 hash function (specified in Annex B.6 in ZigBee Document 053474r17, The ZigBee
2192 Specification, ZigBee Technical Steering Committee (TSC) with a digest size (hashlen) equal to
2193 128 bits).

2194 Installation Code examples:

- 2195 • MMO hash applied to the Installation Code “83FE D340 7A93” produces the key
“CD4FA064773F46941EC986C09963D1A8”.
- 2196

2197 *Note: Least significant byte is 0x83 and Most significant byte is 0x93.*

- 2198 • MMO hash applied to the Installation Code “83FE D340 7A93 9738” produces the key
“A833A77434F3BFBD7A7AB97942149287”.
- 2199

2200 *Note: Least significant byte is 0x83 and Most significant byte is 0x38.*

2201

- 2202 • MMO hash applied to the Installation Code “83FE D340 7A93 9723 A5C6 39FF” produces
the key “58C1828CF7F1C3FE29E7B1024AD84BFA”.
- 2203

2204 *Note: Least significant byte is 0x83 and Most significant byte is 0xFF.*

- 2205 • MMO hash applied to the Installation Code “83FE D340 7A93 9723 A5C6 39B2 6916
2206 D505” produces the key “66B6900981E1EE3CA4206B6B861C02BB”.

2207 Note: Least significant byte is 0x83 and Most significant byte is 0x05.

2209 5.4.8.1.2.1 MMO Hash Code Example

2210 Open source implementations of the MMO Hash based on the Rijndael implementation
2211 are available on the internet at sites like SourceForge and others. The source code is also
2212 available for download from the ZigBee document management system [B6].

2214 5.4.8.2 Multiple Security Credentials

2215

2216 The Key Establishment cluster supports multiple cryptographic suites and, as a result, a device's
2217 security credentials (e.g. certificates) are specific to the cryptographic suite. All clients and
2218 servers SHALL implement and support all crypto-suites. Markets and/or specific deployments
2219 may choose to only use certain crypto-suites at runtime based on their policies.

2220

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.

2226 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].

2239 **5.5.1 Forming the Network (Start-up Sequence)**

2240 Smart Energy devices must form their own network or join an existing network. The
2241 commissioning framework [B2] discusses some of the relevant issues in this procedure.

2242 It is intended that an installer of a Smart Energy device knows if the device is forming a
2243 network or joining an existing network.

2244 If a device is forming a network there is no user interaction required since the form process
2245 can be completed by the device. However there should be some indication to the user or
2246 installer that the network has formed properly. The indication can be implemented in a
2247 number of ways including blinking indicator lights, colored indicator lights, arrays of indicator
2248 lights, text displays, graphic displays, audible indicators such as buzzers and speakers, or
2249 through separate means.

2250 If a device is joining an existing network, it will join the network using the processes
2251 outlined in sub-clause 5.4. Permit joining will have been turned on due to either installer action
2252 or some backchannel mechanism because of user or installer action. It is recommended there
2253 be some indication to the user that the device has joined the network successfully. The
2254 indication can be implemented in a number of ways including blinking indicator lights, colored
2255 indicator lights, arrays of indicator lights, text displays, graphic displays, audible indicators
2256 such as buzzers and speakers, etc.
2257

2258 **5.5.2 Support for Commissioning Modes**

2259 Three different commissioning modes are discussed in [B2]. They are denoted A, E and S-mode.

2260 As discussed above, Smart Energy devices will either automatically form or join a network based
2261 on the processes outlined in sub-clause 5.4.

2262 The pre-installation of start up parameters could be done at manufacturing (which is defined as A
2263 mode), by an installer tool at the dispatching warehouse, or on site (which would then be S
2264 mode). Devices that support this pre-installation must document the methods used for this
2265 preinstallation of parameters to accomplish this process.

2266 Those devices that will join an existing network must support button pushes or simple
2267 documented user interfaces to initiate the joining process. This is in support of E mode
2268 commissioning.
2269

2270 **5.5.3 Commissioning Documentation Best Practices**

2271 To ensure a uniform user experience when commissioning Smart Energy devices, all ZigBee
2272 Smart Energy devices are required to provide documentation with their product that explains how

2273 to perform device commissioning in using a common language set, i.e., “form network”, “join
2274 network”, etc. Please refer to [B2] for further guidance using installation tools and procedures.
2275

2276 **5.5.4 Commissioning Procedure for Different Network Types**

2277 Depending on the type of network being installed, the commissioning procedures may be
2278 slightly different. To ensure interoperability even within these different methods the specific
2279 steps are detailed here.

2280
2281 **5.5.4.1 Commissioning for Neighborhood Area Network or Sub-metering**
2282 Under a neighborhood area network, other meters such as gas or water meters may join
2283 electric meters that form a backbone of the network. The process of joining the network is
2284 separate from the process for device binding where the device billing information is
2285 configured for a particular dwelling unit. It may be desirable to allow the meter to join an
2286 adjacent dwelling unit from a network standpoint to ensure proper connectivity. The
2287 application level will handle the configuration of the billing information later.

- 2288 1 There are two methods for joining such a device onto an existing network:
2289 a The device is commissioned using a tool with the necessary network and security start up
2290 parameters to allow it to rejoin the network as a new device. The device can rejoin any
2291 device in the network since it has all the network information.
2292 b The network has permit joining turned on by an external tool and the device joins this
2293 network and undergoes joining and authentication as any newly joined device.
2294 2 Once joined and authenticated by the security requirements of the existing network, the device
2295 is now a member of the neighborhood area network.
2296 3 At the application level, the particular device ID is associated with a particular dwelling unit
2297 for billing purposes. This information may be associated at the backend database where the
2298 data is collected, or may be sent to the device so it is aware of its association. Note that under
2299 this method, devices may route data through devices in adjacent dwelling units that are part of
2300 the neighborhood area network.

2301
2302 **5.5.4.2 Commissioning for Home Area Network**

2303 Under a home area network, the network consists of devices in a particular dwelling unit
2304 with one or more co-located metering devices or ESI that provides connectivity to the utility
2305 network. Under this scenario, the device within the home may be installed by a trained
2306 installer or by a homeowner. The following steps are completed:

- 2307 1 The Smart Energy network must be informed of the device that is to be joined. This is done
2308 through an out of band means which could include a web login, phone call to a service center,
2309 or handheld tool. Using this methodology the existing network is made aware of the device ID

2310 and security information appropriate for the device (per the Key Establishment Cluster
2311 described in Annex C).

2312 **2** The Smart Energy network is put into permit joining ON for a period of time.

2313 **3** The installer/homeowner is prompted to press a button or complete a menu sequence that tells
2314 the device to attempt to join a network.

2315 **4** The device joins the network and is authenticated using the appropriate security mechanisms
2316 per the Key Establishment Cluster.

2317 **5** An indicator is provided for the installer/homeowner indicating the device has joined a
2318 network and authenticated properly or provides information about improper authentication.

2319 **6** The device can now operate normally on the network.

2320 **5.5.5 ZigBee Smart Energy Joining, Service Discovery, and Device Binding 2321 Requirements**

2322 Commissioning of a device into a ZigBee Smart Energy network should be easy, reliable, and
2323 deterministic. Ideally, a new device could be installed by the home owner or installer
2324 communicating the device install code out of band to the coordinator/trust center and then simply
2325 powering up the device or manually putting the device into a commissioning (auto-join) state.
2326 The device should automatically handle all the steps needed to discover and join the correct PAN
2327 and establish relationships with other devices in the HAN without user intervention. As network
2328 or HAN conditions change, the devices should be able to adapt automatically without user
2329 intervention. ZigBee Smart Energy networks are supposed to last for decades, but once
2330 commissioned, devices should require no user interaction in order to remain part of the ZigBee
2331 PAN.

2332 Devices that are configured with a Startup Parameter of two (un-commissioned) should
2333 automatically begin or make easily available a way to go to Auto-Joining State as described
2334 below. (See sub-clause 5.3.1 for the SE Profile Startup Parameter set.)

2335

2336 **5.5.5.1 PAN Auto-Joining State**

2337 **1** When auto-joining state is initiated, a device shall periodically scan all startup set channels for
2338 networks that are allowing joining. (See sub-clause 5.3.1 for startup set channel description). A
2339 recommended periodic schedule would be:
2340 **a** Immediately when auto-joining state is initiated.
2341 **b** If auto-joining state fails, retry once a minute for the next 15 minutes, jittered by +/- 15
2342 seconds.

- 2343 c If those joining states fail, then retry to join once an hour jittered +/- 30 minutes.
- 2344 2 To find prospective networks to join, the joining node shall send Beacon Request packets on
2345 each channel, dwelling on each channel as specified by the ZigBee PRO specification beacon
2346 response window.
- 2347 3 When a beacon is heard and it has the “Permit Joining” bit set, the device shall attempt to join
2348 that PAN. It is up to the implementation of the device to decide if it wants to survey all
2349 channels and build a list of joinable PANs before attempting a join procedure, or if it should
2350 attempt to join each PAN on a beacon-by-beacon basis. The device shall use its preconfigured
2351 link key (derived from a hash of the installation code) to join the targeted SE PAN.
2352 Exchanging keys in the clear or with well known preconfigured link keys is not allowed.
- 2353 4 If the device joins the network but receives a network key that it cannot decrypt, then it has
2354 likely joined an incorrect PAN and should back out and try the next joinable PAN. This
2355 situation happens most often when the out of band mechanism to communicate the installation
2356 key is flawed, or when more than one PAN is allowing joining. It is permissible to try and join
2357 the same network again, but not recommended that it be done more than three times in
2358 succession. It is expressly not allowed that a device repeat this step more than ten times
2359 without backing off to step two and scanning for other networks to join.
- 2360 5 After the device joins the PAN and is granted ZigBee network key, it must perform service
2361 discovery to find a ZigBee Key Establishment cluster server, then perform ZigBee key
2362 establishment in order to get an APS layer link key.
- 2363 6 If this key establishment fails, it is likely that one side of the exchange is configured with an
2364 invalid certificate or with no certificate at all. It is permissible to retry this step multiple times
2365 in succession, but it is expressly not allowed that a device repeat this step more than ten times in
2366 succession without pausing for a minimum of least fifteen minutes. Since the device was able
2367 to get a network key from the Trust Center, the device must have found the correct PAN to
2368 join, so there is no need to leave the network. A device that does not initiate Key
2369 Establishment with the Trust Center within a reasonable period of time MAY be told to leave
2370 depending on the network operator’s policy. A maximum period of 20 minutes is
2371 recommended.
- 2372 7 Once key establishment succeeds, the device has joined the correct PAN and shall never leave
2373 the PAN without direction from another device in the network (typically an APS *Remove*
2374 Device command from an ESI or ZigBee Network Manager) or direction from the user via the
2375 device user interface. Example user interfaces could be a text menu or a simple button push
2376 sequence. It is strongly recommended that the user interface procedure to get a device to leave
2377 the PAN be explicit and difficult to trigger accidentally. Leave commands received over the air
2378 should only be followed if the command is an APS encrypted APS *Remove* command.
2379 Network layer leave commands should be ignored unless the device is an end device, and the
2380 network leave command originated from the parent device.

2381 **8** A device that leaves a ZigBee network shall discard its network settings and link key, and
2382 revert to its install code, and wait for user input to return or automatically return to auto-join
2383 state step one. The device will require the out-of-band registration process to join a new
2384 network.

2385

2386 **5.5.5.2 Service Discovery State:**

2387 **1** After successfully performing key establishment, devices supporting MDU shall perform
2388 service discovery for the MDU Pairing cluster first and, if found, request pairing information
2389 before continuing discovery for other services. If the MDU Pairing cluster is NOT located on
2390 the Trust Center (ESI), the device providing the cluster MUST be available on the HAN
2391 before other devices are allowed to join. If MDU Pairing Information is returned, the device
2392 should limit service discovery to those devices identified in the returned information. If the
2393 device has joined a ‘normal’ Smart Energy HAN, no Pairing Information will be returned and
2394 the device should perform service discovery with all devices on the network. Refer to 5.6 and
2395 Annex D.13 for further details.

2396 **2** The device should use ZigBee Service Discovery mechanisms to discover other devices on the
2397 network that have services that match with the device's. This would apply to ZigBee Smart
2398 Energy clusters that support asynchronous event commands, like DRLC, Messaging, and Price
2399 clusters. For example, a load control device would use ZigBee service discovery to find ESIs
2400 that support the load control cluster server. (See sub-clause 5.4.5.1 for more details.).

2401 **3** Where a network supports multiple commodity types, the service discovery mechanism may
2402 find multiple instances of certain clusters. In this case, the device should refer to the
2403 *CommodityType* attribute within any Price clusters found, or the *MeteringDeviceType* attribute
2404 within any Metering cluster servers found, in order to determine the appropriate endpoint(s).
2405 Price clusters must always be used where multiple commodities are supported; all clusters
2406 related to the same commodity should be grouped on the same endpoint. If an endpoint
2407 supporting the required commodity is not found, a device shall not use clusters identified as
2408 being for another commodity; the device shall repeat the discovery mechanism at regular
2409 intervals until such time as the correct clusters become available (see point 6 below for
2410 suggested intervals). Clusters for multiple commodities shall not be mixed on the same
2411 endpoint (see section 6.3 for further details).

2412 **4** When a matching service is discovered, the device shall use ZigBee device bind mechanisms
2413 to send a binding request to the matching device endpoint. It is possible that more than one
2414 device with matching services will be discovered. If the device is not an ESI and the ESI are
2415 the matching device(s), the device should send binding requests to all ESI with matching
2416 services. See the “Multiple ESI Application Guidelines” for more details. Hence a device that
2417 wishes to receive unsolicited messages from an ESI on the Messaging Cluster, Price Cluster,
2418 DRLC Cluster, shall issue a bind request to the ESI for each cluster it is interested in.

- 2419 **5** A device that sends a binding request is simply announcing itself to an ESI that it desires
2420 certain sets of information that the ESI may presently have or may obtain in the future, such as
2421 pricing information or DRLC event schedules. The ESIs that receive bind requests are free to
2422 refuse them, but if they refuse the binding request, they must choose another method (an
2423 address table for instance) to note the device's interest. Once a device has issued the binding
2424 request, it does not need to receive a binding response success. If the device receives a
2425 NOT_SUPPORTED (or other non-success code) response to a cluster device bind request, it
2426 should still send binding requests for any remaining clusters that it has not sent already.
- 2427 **6** After the device has discovered and bound to matching services, it has now established an
2428 application layer relationship with all other relevant devices in the HAN (see sub-clause 5.7.2
2429 for details of how to deal with multiple time servers and other duplicated services). That does
2430 not mean that the HAN is static and will not acquire new devices, replace devices, or power on
2431 devices that were not present during the initial discovery phase. To account for a dynamic
2432 HAN, devices shall:
2433 a Repeat the discovery phase on a period of no more than once every three hours and no less
2434 than once every 24 hours.
2435 b Repeat the discovery phase after successfully exiting the Rejoin and Recovery Phase (see
2436 below).
2437 c Optional - Repeat the discovery phase when a device announce broadcast for a full function
2438 device is received. The beginning of the discovery phase should be jittered between 60 and
2439 600 seconds and should be directed only at the device that sent the device announce
2440 broadcast.

2441 **5.5.5.3 Device Steady State**

2442 This is the normal state of the device.

- 2443 **1** A device should make efforts to remain on the correct channel of the PAN and also to keep its
2444 network and application keys in sync with the trust center. It is possible that the device has
2445 missed a key roll or a channel change due to interference or while it has been powered down
2446 or asleep. In order to detect these types of network changes devices shall perform some sort of
2447 APS layer message exchange with an ESI on a regular basis. This is to establish that the device
2448 can still communicate with the ESI using a current network and APS layer key. This exchange
2449 should be performed in accordance with the keep-alive method described in sub-clause
2450 5.4.2.2.3.4. Devices that do not support APS encrypted clusters (Range Extenders for example),
2451 do not need to send APS encrypted packets to the Trust Center, but can send network
2452 encrypted packets instead.
2453
2454 **2** What periodic APS layer message exchange is performed is up to the implementation.
2455 Examples would include:

- 2456 a Reading a mandatory Time cluster attribute (such as *CurrentTime*) on the ESI
2457 (recommended). This should work for all ESI.
- 2458 b Reading the current consumption attribute on the Metering cluster (if the ESI supports the
2459 Metering server).
- 2460 c Requesting next pricing info from the ESI (if the ESI supports the Price cluster server).
- 2461 **3** If the device attempts to perform the periodic message exchange and it fails for any reason, the
2462 device should note the failure and retry another exchange later. If after no more than twenty-
2463 four hours of retries have failed, the device shall go into the Rejoin and Recovery Phase. It is
2464 left to the implementation to decide how many retries should occur within the 24 hour period.
2465 It is also permissible for the device to enter the Rejoin and Recovery Phase earlier than 24
2466 hours based on number of failed retries or other factors.
- 2467 **4** Sleepy end devices are not required to periodically communicate with an ESI. Instead they
2468 should periodically poll their parents and if no parent is found after a suitable period find and
2469 rejoin to a new parent. If no parent is found on the original channel, the end device should
2470 enter the Rejoin and Recovery phase described below to find a new parent.

2471

2472 5.5.5.4 Rejoin and Recovery State

2473 A device in Rejoin and Recovery Phase is trying to get in sync with its PAN.

- 2474 **1** The device in R&R Phase shall first attempt a ZigBee secure rejoin procedure on its current
2475 channel. If the secure rejoin procedure succeeds, the device should revert to its steady state
2476 behavior.
- 2477 **2** If the secure rejoin procedure fails, it shall attempt to do a trust center rejoin procedure on its
2478 current channel.
- 2479 **3** If the trust center rejoin procedure fails, it may optionally retry steps one and two up to three
2480 times.
- 2481 **4** If all attempts to rejoin on the current channel fail, the device shall scan all other channels for
2482 its PAN by issuing beacon requests. Note that the PAN ID may have changed and the device
2483 shall compare with the extended PAN ID in the beacon and not the short PAN ID.
- 2484 **5** If the device finds an extended PAN ID match in a received beacon, it shall repeat steps one
2485 and two on the new channel.
- 2486 **6** If the rejoin (and optional retries) fail on the new channel, the device shall continue scanning
2487 all remaining channels for its PAN.

- 2488 7 If no correct PANs are discovered on any channel, the device shall return to its original
2489 channel to wait for the next R&R attempt.
- 2490 8 If all rejoin attempts on all channels fail, the device shall return to its original channel to wait
2491 for the next R&R attempt. This means that the device is back on the original PAN channel, is
2492 still a member of the original PAN, (it has not left the network, and has not discarded any PAN
2493 information or security keys), and is simply waiting for the rest of the PAN to appear or to time
2494 out and begin another R&R attempt.
- 2495 9 If while waiting for the next R&R attempt, the device receives an APS encrypted message
2496 from an ESI and is encrypted with the device's current network and APS layer key, the device
2497 shall leave the R&R phase and proceed to the steady state phase.
- 2498 10 While in the R&R phase, the device shall retry steps 1-8 periodically, at least once every 24
2499 hours. Sleepy end devices may use a longer period. After four failed rejoin attempts, devices
2500 should not try to rejoin any faster than once per hour, with a jitter of +/- 30 minutes.

2501 5.5.5.5 ESI Specific Considerations

- 2502 1 ESI that are not the PAN coordinator, trust center, or network manager shall perform the
2503 steady state phase and rejoin and recovery phase as described above.
- 2504 2 ESI shall support at a minimum, through bindings or other means, at least five separate
2505 devices, with enough resources for each device to bind to all of the relevant clusters that the
2506 devices may request bindings to. For example, if the ESI supports five smart energy clusters
2507 that devices may send binding requests for, the ESI must support twenty five binding
2508 relationships, as well as five sets of device ids and security keys
- 2509 3 It is strongly recommended that ESI operators remove inactive or deprecated devices from the
2510 HAN as well as ESI key and binding tables before adding new devices in order to make room
2511 for the new device(s). This use case is an example of a device replacement in the HAN.
- 2512 4 The Trust Center shall never issue an APS *Remove* command without an explicit request from
2513 another device on the network or from the head-end network management system.
- 2514 5 When a new device is registered with the ESI, the ESI may not have enough resources to
2515 support it. If the ESI is low on resources, it should notify the installer or ESI administrator (this
2516 could be via user interface, or backhaul/ backchannel communication for example.) The ESI
2517 shall not automatically remove other devices in order to free up resources for the new device
2518 without explicit approval from the installer or ESI administrator.
- 2519 6 If a device joins the PAN, but does not successfully perform Key Establishment, a trust center
2520 may remove the device. This shall only be done after more than 1 hour has elapsed since the
2521 device's initial join. This shall be sent directly to the router, or to the parent of an end device. A

2522 child that receives a NWK leave from its parent when it does not have an authorized link key
2523 (i.e. not performed key establishment successfully) shall not ignore the leave.

2524 **7** It is permissible and encouraged that an ESI perform its own service discovery procedure after
2525 power up and on a periodic basis. The ESI may independently create its own bindings to
2526 devices with matching services. This may in some cases establish application layer
2527 relationships faster than waiting for devices to request bindings by themselves. As specified in
2528 sub-clause 5.4.5.1, an ESI that does not support bindings shall perform its own service
2529 discovery.

2530

2531 **5.6 Federated Trust Center Application Guidelines**

2532 *Note: The Federated Trust Center feature in this revision of this specification is provisional
2533 and is not certifiable. This feature set may change before reaching certifiable status in a future
2534 revision of this specification.*

2535 When considering Smart Energy systems to support multiple dwelling units (MDUs) such as
2536 apartment blocks, one solution is to use a single ZigBee mesh network (NAN) to support the
2537 entire building. In this scenario, there will be a single Trust Center (ESI), also known as a
2538 *Federated Trust Center*, that will control this entire network.

2539 The devices associated with a particular apartment within the MDU will form a ‘virtual HAN’,
2540 and will consist of devices within the apartment together with the associated meter(s) which may
2541 be located in a central meter room or elsewhere within the building. Thus the overall network for
2542 the entire building will be made up of a number of groups of devices.

2543 The Federated Trust Center will need to be pre-loaded with information indicating the devices
2544 that will form the ‘virtual HAN’ for each apartment. Additional functionality will be required so
2545 that a device joining this NAN can be made aware of the other devices that will constitute the
2546 virtual network for the relevant apartment; the device will then limit its subsequent service
2547 discovery to that list of devices.

2548 Once a device has joined the NAN, and has successfully completed key establishment with the
2549 Trust Center, it shall request Pairing Information from the Trust Center. Once Pairing
2550 Information is returned, the device shall limit service discovery to those devices identified in the
2551 response. It should be noted that, in this scenario, service discovery will be performed using
2552 unicast commands rather than broadcast.

2553 Devices may join and leave the ‘virtual HAN’ from time to time. To allow for this possible
2554 reorganization, each device within the NAN shall periodically ask the Federated Trust Center

2555 whether new Pairing Information is available. The typical period is once every 24 hours. A
2556 *Version* field within the relevant commands allows updated information to be identified.

2557 For further information, refer to the MDU Pairing cluster definition in Annex D.13.

2558 **5.7 Multiple ESI Application Guidelines**

2559 **5.7.1 Overview**

2560 The ZigBee Smart Energy Profile allows for the use of multiple ESIs in a HAN. This feature is
2561 desirable from a reliability perspective, plus opens opportunities for vendors to innovate and
2562 provide additional services and functionality. Multiple ESIs does not mean multiple Trust Centers,
2563 only a single Trust Center is supported in a HAN.

2564 Clients may assume that all SE messages/directives (Demand Response events, price
2565 publishing, messaging) are created by the same entity, e.g., utility or energy management entity,
2566 or set of coordinated entities. These messages can be sent to devices via one or more transport
2567 mechanisms (in the HAN, this means the same message may be sent from multiple ESIs). A
2568 message with a specific ID typically will be unique within the system, even though a device may
2569 receive this message more than once. However, in a HAN with multiple, uncoordinated
2570 commodity service providers (e.g., gas vs. water, household electricity vs. PEV electricity), there
2571 is a possibility that different, unique events will have conflicting event IDs. Since it is expected
2572 that ID conflicts for events occurring at similar points of time will be rare, clients may ignore the
2573 issue and always assume that conflicting event IDs are duplicates. More complex clients may
2574 choose to better track events by service provider, commodity type, etc.

2575 *Note: The functionality in this section does not describe the operation of multiple ESIs within a*
2576 *MDU/NAN. Refer to section 5.6 for details of Federated Trust Center functionality.*

2577 **5.7.2 Device Behavior**

2578 **5.7.2.1 Service Discovery in Multi ESI Environments**

2579 A device should make itself aware of any and all ESIs in a SE HAN using service discovery. It
2580 shall perform this service discovery upon joining a network, power up (and network rejoin), and
2581 periodically. The typical period is once every 24 hours. A device that discovers an ESI with
2582 matching services shall create bindings on the ESI so that the ESI will register the device and
2583 send it appropriate SE commands. Devices which do not bind in a multiple ESI network are
2584 expected to poll the ESIs. ESIs that are not rediscovered over the period of multiple
2585 discovery cycles may be forgotten by the device.

2586 A device that discovers more than one ESI should determine a single ESI as an authoritative
2587 time source. To do so, it should use the Time cluster Master, Synchronized, and Superseding bits.

2588 A Time server with the Superseding bit set will always take precedence over a Time server
2589 without that bit set, including ones that have the Master bit set. A new ESI going into a faulty
2590 installation can set the Superseding bit and take over the network's Time synchronization.
2591 However, it is not required for SE 1.x to have this bit set if the new ESI does not want to
2592 forcefully take over the Time server role. This bit is set independently of the other three
2593 *TimeStatus* bits (Master, Synchronized, MasterZoneDst).

2594

2595 **5.7.2.2 Determining the Most Authoritative Time Source**

2596 Devices shall synchronize to a Time server with the highest rank according to the following
2597 rules, listed in order of precedence:

2598 **1** A server with the Superseding and Master bits set shall be chosen over a server with only the
2599 Master bit set.

2600 **2** A server with the Master bit set shall be chosen over a server without the bit set.

2601 **3** The server with the lower short address shall be chosen (note that this means a coordinator
2602 with the Superseding and Master bit set will always be chosen as the network time server).

2603 **4** A Time server with neither the Master nor Synchronized bits set should not be chosen as the
2604 network time server.

2605 **5.7.2.3 Periodic Time Source Checking During Normal Operation**

2606 During normal operation (the most authoritative time source is found and it has valid time),
2607 clients periodically repeat the time source scan to pick up new, more authoritative time sources,
2608 as per the following rules:

2609 **1** Non-sleepy clients shall locate the most authoritative time source at least once every 24 hours.

2610 **2** Sleepy devices should locate the most authoritative time source at least once every 24 hours.

2611 **3** Clients shall scan for time sources after rebooting, and after joining or rejoining the network.

2612 **5.7.2.4 Invalid Time and Interim Time Sources**

2613 Although the rules above are used to find the most authoritative time source for the network,
2614 there are conditions where what would normally be the most authoritative source is
2615 temporarily unable to provide valid time. In this situation, regardless of whether it is encountered
2616 as part of the original time discovery or the periodic rediscovery, devices obey the following
2617 rules:

- 2618 1 If a server is temporarily unable to provide valid UTC, it shall report all time attributes (e.g.,
2619 UTC, local time) as 0xFFFFFFFF. It should leave the Superseding, Master, and Synchronized
2620 bits set as if it did have valid time.
- 2621 2 If the most authoritative time source for the network has invalid time, clients should
2622 temporarily use the Time server of next highest rank, but shall periodically look for the more
2623 authoritative server(s) to obtain valid time. Non-sleepy devices shall check at least once every
2624 15 minutes. Sleepy devices may check as often as their power budget will allow.
- 2625 3 When a more authoritative time source with valid time is found, clients shall immediately
2626 switch to using that source's time basis.

2627 5.7.2.5 Handling SE Commands from Multiple ESIs

2628 When a device creates bindings on multiple ESIs, it may receive SE commands from those
2629 ESIs. Simple device logic such as assuming all commands came from the PAN coordinator is not
2630 appropriate. The following rules describe the desirable device behavior.

- 2631 1 When a device receives an event (Demand Response, Price, Messaging) any time reference in
2632 the message should be viewed in context with the time reference of the most authoritative ESI
2633 time server.
- 2634 2 When a device receives duplicate events (same event ID) from multiple ESIs, it shall send an
2635 event response to each ESI. Future duplicate events from the same ESI(s) shall be either
2636 “ignored” by sending no response at all or with a default response containing a success status
2637 code.
- 2638 3 Conflicting events with the same event ID from different ESIs will be resolved by the device in
2639 the same manner as if they came from a single ESI.
- 2640 4 When a device has an asynchronous follow up event response it should send the response to
2641 the ESIs that created the condition. If the event was received from more than one ESI, the
2642 device shall send the asynchronous event response to all ESIs from which it received the event.

2643 5.7.2.6 Handling Multiple Uncoordinated Back-end Systems

2644 When multiple, uncoordinated service providers deploy ESIs in a HAN, it is possible that
2645 different back-ends/ESIs will have a different notion of time. However, only one ESI will
2646 be the authoritative time source for the HAN. In this scenario, an ESI may require a mechanism
2647 to ensure that its events are executed on its time basis, even if it is not the authoritative time
2648 source. ESIs that require this behavior may make use of the following application guidelines:

- 2649 1 An ESI may implement the Time cluster client and determine the most authoritative time
2650 source for the HAN, using the rules defined above.

- 2651 **2** If the ESI is not the authoritative time source, it may synchronize its clock, or apply
2652 differentials to the start time of its events, to ensure that clients execute those events on the
2653 intended schedule.
- 2654 **3** If the more authoritative time server disappears and is not seen for 24 hours, clients may
2655 assume that the server has left the network, and resume normal operation using the most
2656 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 When forming a new network, or scanning to join a network, Smart Energy devices should
2660 do channel scans using the following preferred channels before scanning the rest of the
2661 channels in order to avoid the most commonly used WiFi channels. This is to improve the user
2662 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 Broadcasts are strongly discouraged for Smart Energy devices. Devices are limited to a
2667 maximum broadcast frequency of one broadcast per second and strongly encouraged to exercise
2668 broadcasts much less frequently.

2669

5.8.3 Frequency Agility

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

2675

5.8.4 Key Updates

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

2680

5.9 Coexistence and Interoperability with HA Devices

2681 It is desirable to allow interoperability of HA and Smart Energy devices where practical.
2682 However, it is undesirable to publicly share keys during the joining process or share private
2683 information over a less secure network. HA devices that only provide functionality for
2684 receiving network keys in the clear during a join process cannot be used in a Smart Energy
2685 network. HA devices may also be extended with Smart Energy clusters providing they support
2686 the use of Link Keys and the Smart Energy security models. If so, they can be certified as
2687 HA and Smart Energy capable allowing those devices to operate either in an HA network or a
2688 Smart Energy network.

2689

5.10 Device Descriptions

2690 Device descriptions specified in this profile are summarized in Table 5-14 along with their
2691 respective Device IDs. The devices are organized according to the end application areas they
2692 address. A product that conforms to this specification shall implement at least one of these
2693 device descriptions and shall also include the device descriptions corresponding to all
2694 applications implemented on the product where a standard device description is specified in this
2695 profile. For example, if a product implements both a thermostat and an In-Home Display, then
2696 the thermostat and In-Home Display device descriptions must both be supported.

2697 This list will be added to in future versions of the profile as new clusters are developed to meet
2698 the needs of manufacturers. The reserved values shall not be used until the profile defines them.
2699 Manufacturer-specific device descriptions shall reside on a separate endpoint and use a private
2700 profile ID.

2701

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

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

2702	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">Prepayment Terminal</td><td style="padding: 2px;">0x0506</td></tr> <tr> <td style="padding: 2px;">Physical Device</td><td style="padding: 2px;">0x0507</td></tr> <tr> <td style="padding: 2px;">Remote Communications Device</td><td style="padding: 2px;">0x0508</td></tr> <tr> <td style="padding: 2px;">Reserved</td><td style="padding: 2px;">0x0509 – 0x5FF</td></tr> </table>	Prepayment Terminal	0x0506	Physical Device	0x0507	Remote Communications Device	0x0508	Reserved	0x0509 – 0x5FF
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 handling. 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.

-
- 2728 • When receiving requests for accessing unsupported attributes, devices shall respond using
2729 the ZCL default response with a status code of UNSUPPORTED_ATTRIBUTE.
- 2730 • When receiving requests for accessing attributes where there is an additional access control
2731 mechanism and access is denied, devices shall respond using the ZCL default response with
2732 a status code of NOT_AUTHORIZED. Any additional access control mechanism is optional
2733 and the specification of the additional access control mechanism is out of scope of this
2734 specification.

2735 Please refer to [B1] for additional status codes support in the ZCL default response.

2736 **5.12 Cluster List and IDs**

2737 The clusters used in this profile are listed in Table 5.15. The clusters are listed according to
2738 the functional domain they belong to in the ZCL and indicate the additional new Smart
2739 Energy clusters. The existing corresponding ZCL General cluster identifiers can be found in the
2740 ZCL [B1].

2741 The functionality made available by all supported clusters shall be that given in their ZCL
2742 specifications except where a device description in this profile includes further specification,
2743 clarification or restriction as needed for a particular device.

2744 Most clusters include optional attributes. The application designer must be aware that optional
2745 attributes might not be implemented on a particular device. All Smart Energy devices must
2746 discover and deal with unsupported attributes on other devices.

2747 It is expected that clusters will continue to be developed in the ZCL that will be useful in this
2748 profile. In many cases, new clusters will be organized into new device descriptions that are
2749 separate from those currently defined. There may also be situations where it makes sense to add
2750 clusters as optional elements of existing device descriptions.

2751 Manufacturer-specific clusters may be added to any device description in this profile as long
2752 as they follow the specifications given in the ZCL [B1].

2753

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

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

2754
2755

2756 5.12.1 ZCL General Clusters

2757 Except for the Key Establishment Cluster, which is covered in Annex C, please refer to the
2758 ZCL Cluster Specification [B1] for the General Cluster descriptions.

2759 5.12.1.1 ZCL Time Cluster and Time Synchronization

2760 The Smart Energy profile requires time synchronization between devices to properly
2761 support the coordination of Demand Response/Load Control events, Price changes, and the
2762 collection of metered data. In order to simplify the understanding of time, the Smart
2763 Energy profile will leverage UTC as the common time base. To this end a new ZCL
2764 attribute data type, UTCTime is included and its definition can be found in Annex A.

2765 It is desired for the processes for synchronizing time to be as network friendly as possible to
2766 eliminate excessive traffic. To support this, time accuracy on Client devices shall be within +/-
2767 1 minute of the server device (ESI) per 24 hour period. The Client devices shall design a
2768 clock accuracy that never requires more than one time synchronization event per 24 hour period.
2769 The exception to this is when devices need to rejoin or re-register on the network. Again, the
2770 desire is to keep time synchronization traffic to a minimum.

2771 Further, implementers must be aware that network communication delays will cause minor
2772 differences in time between devices. The Smart Energy profile expectations are that this will
2773 be a minor issue given the use cases it's fulfilling. It will not nor does it recommend

2774 implementers develop an NTP or equivalent scheme to compensate for network delays. These
2775 methods are viewed as having the potential to cause excessive network communications.

2776 **5.12.1.2 Transaction Sequence Numbers**

2777 The normal usage of Transaction Sequence Numbers is defined in [B1] 2.3.1.3. The following
2778 clarifications shall be noted with respect to the usage of Transaction Sequence Numbers within
2779 the Smart Energy Standard:

2780 **1** Where a ‘Publish’ command is sent in response to the receipt of an associated ‘Get’
2781 command, the ‘Publish’ command shall use the Transaction Sequence Number received in
2782 the ‘Get’ command

2783 **2** Where there is more than one response to a request, the Transaction Sequence Number from
2784 the request shall be used in all responses to that specific request

2785 **3** Where a ‘Publish’ command is sent unsolicited, the server is free to choose any value of
2786 Transaction Sequence Number

2787 A Transaction Sequence Number is not to be used to reject a command, only to correlate
2788 messages.

2789

2790 **5.13 Coexistence with devices using other Profiles**

2791 *Note: The Profile Coexistence feature in this revision of this specification is provisional and is
2792 not certifiable. This feature set may change before reaching certifiable status in a future revision
2793 of this specification.*

2794 Devices that do not implement Smart Energy 1.x clusters may be allowed on to the ZigBee
2795 network at the discretion of the network owner. Additional router devices have the benefit of
2796 growing the range and reliability of the ZigBee network, and offering additional services not
2797 offered by the Smart Energy profile. This also benefits the user as it means fewer ZigBee
2798 networks to manage.

2799 This section describes only how Smart Energy devices and devices from other profiles will
2800 interact in a general sense to join the network and communicate. It does not describe how
2801 specific clusters from one profile may be used to communicate with clusters on a different
2802 profile.

2803 All devices in the Smart Energy protocol are required to adhere to these requirements regardless
2804 of whether the network owner chooses to allow devices from other profiles on their network.

2805 **5.13.1 Requirements for the Smart Energy Trust Center**

- 2806 **1** The Trust Center shall only allow devices onto the Smart Energy network that utilize
2807 installation code based link keys. This is the normal behavior per the Smart Energy profile.
- 2808 **2** If a Trust Center does not want to allow non-Smart Energy devices onto the network, it is
2809 recommended that the TC should check whether a joining device supports the Key
2810 Establishment cluster and, if not, request the joining device to leave the network.
- 2811 **3** It is recommended that Smart Energy devices that fail CBKE should remove themselves
2812 from the network.
- 2813 **4** Trust Centers that support both Smart Energy and non-Smart Energy devices should not
2814 remove non-Smart Energy devices that do not attempt CBKE.
- 2815 **5** If the Trust Center sends NWK key updates via unicast, then it shall still include devices
2816 from other profiles that have not performed CBKE.
- 2817 **6** Per normal requirements, the Trust Center shall not grant partner link keys to devices that
2818 have not performed CBKE.
- 2819 **7** The Trust Center shall use a device's install code key to transport the NWK key if a Trust
2820 Center rejoin is performed by a device from another profile.
2821

2822 **5.13.2 Requirements for Smart Energy Devices**

- 2823 **1** Smart Energy endpoints that receive Smart Energy cluster messages using a profile other
2824 than the Smart Energy profile shall drop the messages.
- 2825 **2** Smart Energy endpoints must continue to follow the cluster security requirements specified
2826 in Table 5-13 and drop Smart Energy messages that do not have APS security when it is
2827 required (refer to section 5.4.6 for further details).
- 2828 **3** Smart Energy endpoints that also implement clusters from another profile shall not use their
2829 install code key for any encryption or decryption once they establish a CBKE link key.
- 2830 **4** Smart Energy endpoints shall not refer to a Smart Energy-related cluster on a non-Smart-
2831 Energy endpoint. Note that this does not preclude a non-Smart-Energy version of a cluster
2832 existing on a HAN as well as a Smart Energy-related version.

2833

2834 **5.13.3 Requirements for Devices from other Profiles**

2835 **5.13.3.1 Joining**

- 2836 1 Devices from other profiles must implement install code based link keys. This includes
2837 proper documentation of the install code via packaging and instructions provided to the
2838 installer. Refer to section 5.4.8 for further details.
- 2839 2 The Smart Energy network must be informed of the device that is to be joined. This is done
2840 through an out of band means which could include a web login, phone call to a service
2841 center, or handheld tool. Using this methodology the existing network is made aware of the
2842 device ID and Install Code/Link Key for the device.
- 2843 3 The Smart Energy network is put into permit joining ON for a period of time.
- 2844 4 The installer/homeowner is prompted to press a button or complete a menu sequence that
2845 tells the device to attempt to join a network.
- 2846 5 The device joins the network.
- 2847 6 An indicator is provided for the installer/homeowner indicating the device has joined a
2848 network.
- 2849 7 The device can now operate normally on the network according to the rules of the relevant
2850 profile.
- 2851 8 Devices from other profiles shall have no expectations they may use their link key with the
2852 Trust Center other than to encrypt/decrypt APS command frames. Their link key will be
2853 treated like an unauthorized link key. The Trust Center shall not accept APS data frames, for
2854 Smart Energy clusters requiring APS encryption, where such data frames are encrypted
2855 using an unauthorized link key.
- 2856 9 Currently there is no provision in this specification for a device from another profile to
2857 obtain an authorized link key.

2858

2859 **5.13.3.2 Rejoining**

- 2860 Devices from other profiles may perform a secure or trust center rejoin to get back on the
2861 network. A trust center rejoin shall use the device's install code key.

2862

2863 6 Device Specifications

2864 6.1 Common Clusters

2865 Support for certain clusters is required on all products supporting this profile. At least one
 2866 instance of the clusters shown in Table 6-1 shall exist on a product supporting this profile.
 2867 Individual device descriptions may place further restrictions on support of the optional clusters
 2868 shown here. ZCL clusters not listed may be implemented on the Smart Energy endpoint.
 2869 Manufacturers may extend the SE profile as specified in the ZCL specification [B1].

2870 **Table 6-1 – Clusters Common to All Devices**

Server Side	Client Side
Mandatory	
Basic	<i>None</i>
Key Establishment	Key Establishment
Optional	
Power Configuration	None
Commissioning	Commissioning
Identify	<i>None</i>
Events	
OTA Upgrade	OTA Upgrade

2871
 2872

2873 **6.1.1 Optional Support for Clusters with Reporting Capability**
 2874 Some clusters support the ability to report changes to the value of particular attributes. These
 2875 reports are typically received by the client side of the cluster. All devices in this profile may
 2876 support any cluster that receives attribute reports.

2877 **6.1.2 Manufacturer-Specific Clusters**

2878 The ZCL provides a range of cluster IDs that are reserved for manufacturer-specific
 2879 clusters. Manufacturer-specific clusters that conform to the requirements given in the ZCL may
 2880 be added to any device description specified in this profile.

2881 **6.1.3 Cluster Usage Restrictions**

2882 None.

2883 **6.1.4 Identify Cluster Best Practices**

2884 To help aid in locating devices, it's strongly recommended that all devices utilize the Identify
2885 Cluster and a visual or audible indicator. In situations in which a device can't supply a visual
2886 or audible indicator, the device should include a visible label with the appropriate information
2887 to help identify the device.

2888 **6.1.5 Inter-PAN Communication**

2889 Inter-PAN access to Smart Energy devices shall be limited to specific clusters and commands.
2890 Please refer to Annex B for further details.

2891 **6.2 Feature and Function Description**

2892 Each device must support a certain set of features and functions. Table 6-2 below is used to
2893 specify the mandatory and optional features and functions for Smart Energy devices. This
2894 chapter contains a description of what must be supported if the feature or function is supported
2895 by the device. The mandatory or optional configuration for each device is described in the
2896 upcoming chapters:

2897 **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

2898
2899
2900 **Join (End Devices and Routers):**
2901 As described in sub-clauses 5.4 and 5.5.

- 2902 **Form Network (Coordinator):**
2903 As described in sub-clauses 5.4 and 5.5.
- 2904 **Allow Others to Join Network (Router and Coordinator Only):**
2905 As described in sub-clauses 5.4 and 5.5.
- 2906 **Restore to Factory Fresh Settings:**
2907 The Device shall provide a way to restore Factory Settings.
- 2908 **Pair Devices (End Device Bind Request):**
2909 Whenever possible, the device should provide a way for the user to issue an End Device Bind
2910 Request.
- 2911 **Bind Manager (End Device Bind Response – Coordinator only):**
2912 The coordinator device shall be capable of issuing an End Device Bind Response.
- 2913 **Enable Identify Mode:**
2914 Whenever possible, the device should provide a way for the user to enable Identify for 60
2915 seconds.
- 2916 **Service Discovery (Match Descriptor Request):**
2917 Whenever possible, the device should provide a way for device to send a match descriptor
2918 request, receive match descriptor responses and utilize them for commissioning the device.
- 2919 **ZDP Bind Response:**
2920 The device shall be able to receive a ZDP Bind Request and respond correctly with a ZDP
2921 Bind Response.
- 2922 **ZDP Unbind Response:**
2923 The device shall be able to receive a ZDP Unbind Request and respond correctly with a ZDP
2924 Unbind Response.
- 2925 **End Device Annce/Device Annce:**
2926 The device shall Send End Device Annce / Send Device upon joining and re- joining a
2927 network.
- 2928 **Service Discovery Response:**
2929 The Device shall be able to receive a Match descriptor request, and respond with a match
2930 descriptor response correctly.
- 2931 **Allow Smart Energy Devices to Join the Network:**
2932 The Device shall allow other Smart Energy devices to join the network.
- 2933 **High Security Supported:** No

2934 **Enhanced Inter-PAN Communication:**

2935 The device may support Enhanced Inter-PAN Communications as described in Annex B

2936 **6.3 Smart Energy Devices**

2937 A physical device may support one or more logical Smart Energy devices. The supported
2938 clusters of a logical Smart Energy device shall reside on a single endpoint, with the
2939 exception of the common clusters listed in Table 6-1, which may reside on a separate endpoint
2940 using the Physical Device identifier. Each logical Smart Energy device on a single physical
2941 device shall reside on its own separate endpoint.

2942 SE devices shall use the device and service discovery mechanisms specified in the ZigBee
2943 specification [B3] to find the services required. Devices shall support discovery of single and
2944 multiple endpoints on a single physical device. In the case where multiple devices of the same
2945 type are discovered, SE cluster attributes should be read to determine the type of service
2946 provided. For example, if multiple Metering devices are discovered the *MeteringDeviceType*
2947 attribute provides a label for identifying the type of metering device present. Similarly, if
2948 multiple ESIs are found the *CommodityType* attribute shall be read to determine the fuel type
2949 of that ESI.

2950 **6.3.1 Energy Service Interface**

2951 The Energy Service Interface connects the energy supply company communication
2952 network to the metering and energy management devices within the home. It routes messages
2953 to and from the relevant end points. It may be installed within a meter, thermostat, or In-
2954 Home Display, or may be a standalone device, and it will contain another non-ZigBee
2955 communication module (e.g. power-line carrier, RF, GPRS, broadband Internet connection).

2956 **6.3.1.1 Supported Clusters**

2957 In addition to those specified in Table 6-1, the Energy Service Interface device shall support the
2958 clusters listed in Table 6-3. If a SE cluster is not listed as mandatory or optional in the
2959 following table or in the common table, then that cluster shall be prohibited on an ESI device
2960 endpoint.

2961 **Table 6-3 – Clusters Supported by the Energy Service Interface**

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

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

2962

2963

2964 **6.3.1.2 Supported Features and Functions**

2965 The Energy Service Interface device shall have the features and functions listed in Table 6-2.

2966 **6.3.2 Metering Device**

2967 The Metering end device is a meter (electricity, gas, water, heat, etc.) that is fitted with a ZigBee
 2968 device. Depending on what is being metered, the device may be capable of immediate
 2969 (requested) reads or it will autonomously send readings periodically. A Metering end device
 2970 may also be capable of communicating certain status indicators (e.g. battery low, tamper
 2971 detected).

2972 **6.3.2.1 Supported Clusters**

2973 In addition to those specified in Table 6-1, the Metering Device shall support the clusters listed
 2974 in Table 6-4. If a SE cluster is not listed as mandatory or optional in the following table or in the
 2975 common table, then that cluster shall be prohibited on a Metering device endpoint.

2976 **Table 6-4 – Clusters Supported by the Metering Device**

Server Side	Client Side
Mandatory	
Metering	
Optional	
	Time

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

2981 **6.3.3 In-Home Display Device**

2982 The In-Home Display device will relay energy consumption data to the user by way of a
 2983 graphical or text display. The display may or may not be an interactive device. At a minimum at
 2984 least one of the following should be displayed: current energy usage, a history over
 2985 selectable periods, pricing information, or text messages. As an interactive device, it can be
 2986 used for returning simple messages for interpretation by the recipient (e.g. "Button A was
 2987 pressed").

2988 The display may also show critical pricing information to advise the customer when peaks are
 2989 due to occur so that they can take appropriate action.

2990 **6.3.3.1 Supported Clusters**

2991 In addition to those specified in Table 6-1, the In-Home Display device shall support the clusters
 2992 listed in Table 6-5. If a SE cluster is not listed as mandatory or optional in the following table or
 2993 in the common table, then that cluster shall be prohibited on an In-Home Display device endpoint.

2994 **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

2995
 2996 An In-Home Display shall implement at least one of the optional client clusters listed.

2997 **6.3.3.2 Supported Features and Functions**

2998 The In-Home Display device shall have the features and functions listed in Table 6-2.

2999

3000 **6.3.4 Programmable Communicating Thermostat (PCT) Device**

3001 The PCT device shall provide the capability to control the premises heating and cooling systems.

3003 **6.3.4.1 Supported Clusters**

3004 In addition to those specified in Table 6-1, the PCT device shall support the clusters listed in
3005 Table 6-6. If a SE cluster is not listed as mandatory or optional in the following table or in the
3006 common table, then that cluster shall be prohibited on a PCT device endpoint.

3007 **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

3008

3009 **6.3.4.2 Supported Features and Functions**

3010 The PCT device shall have the features and functions listed in Table 6-2.

3011 **6.3.5 Load Control Device**

3012 The Load Control device is capable of receiving Demand Response and Load Control events
3013 to manage consumption on a range of devices. Example devices are water heaters, exterior
3014 lighting, and pool pumps.

3015 [6.3.5.1 Supported Clusters](#)

3016 In addition to those specified in Table 6-1, the Load Control device shall support the clusters
 3017 listed in Table 6-7.

3018 **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

3019

3020 [6.3.5.2 Supported Features and Functions](#)

3021 The Load Control Device shall support the features and functions listed in Table 6-2.

3022 **6.3.6 Range Extender Device**

3023 The Range Extender is a simple device that acts as a router for other devices. The Range
 3024 Extender device shall not be a ZigBee end device. A product that implements the Range
 3025 Extender device shall not implement any other devices defined in this profile. This device shall
 3026 only be used if the product is not intended to have any other application, or if a private
 3027 application is implemented that has not been addressed by this profile.

3028 [6.3.6.1 Supported Clusters](#)

3029 The Range Extender device shall support the mandatory common clusters listed in Table 6-1.

3030 [6.3.6.2 Supported Features and Functions](#)

3031 The Range Extender device shall have the features and functions listed in Table 6-2.

3032 **6.3.7 Smart Appliance Device**

3033 Smart Appliance devices on the ZigBee network can participate in energy management
 3034 activities. Examples of these are when Utilities initiate a demand response or pricing event, or

3035 the appliance actively informs customers via in-home displays of when or how energy is
3036 being used. In the latter case, scenarios include:

- 3037 • Washer switching to cold water during periods of higher energy costs.
3038 • Washer/Dryer/Oven/Hot Water Heater reporting cycle status.
3039 • Over temperature conditions in Freezers and Refrigerators.

3040 **6.3.7.1 Supported Clusters**

3041 In addition to those specified in Table 6-1 the Smart Appliance device shall support the
3042 clusters listed in Table 6-8. If a SE cluster is not listed as mandatory or optional in the following
3043 table or in the common table, then that cluster shall be prohibited on a Smart Appliance device
3044 endpoint.

3045 **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

3046
3047
3048 **6.3.7.2 Supported Features and Functions**

3049 The Smart Appliance device shall have the features and functions listed in Table 6-2.

3050
3051 **6.3.8 Prepayment Terminal Device**
3052 The Prepayment Terminal device will allow utility customers or other users (e.g. sub-metered
3053 tenants) to pay for consumption in discrete increments rather than establishing a traditional

3054 billing agreement. The Prepayment Terminal device will accept payment (e.g. credit card, code
 3055 entry), display remaining balances, and alert the user of a balance approaching zero, and may
 3056 perform some or all of the other functions described in sub-clause 6.3.3 “In-Home Display
 3057 Device”.

3058 **6.3.8.1 Supported Clusters**

3059 In addition to those specified in Table 6-1, the Prepayment Terminal device shall support the
 3060 clusters listed in Table 6-9. If a SE cluster is not listed as mandatory or optional in the following
 3061 table or in the common table, then that cluster shall be prohibited on a Prepayment Terminal
 3062 device endpoint.

3063 **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

3064
 3065 **6.3.8.2 Supported Features and Functions**
 3066 The Prepayment Terminal device shall have the features and functions listed in Table 6-2.

3067 **6.3.9 Physical Device**

3068 The Physical Device type will identify a supplemental (or sole) endpoint on which the clusters
 3069 related to a physical product may reside. The endpoint shall not contain any cluster related
 3070 to any individual logical SE device on the physical product. A product is allowed to have a
 3071 Physical Device as its sole SE endpoint. A Physical Device must be capable of providing other
 3072 SE device endpoints to be a certified SE product.

- 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

3098

3099 **Annex A CANDIDATE ZCL MATERIAL FOR USE WITH THIS PROFILE**

3100 The candidate material in this annex, when approved, will be merged into the Foundation
3101 document of the ZigBee Cluster Library (ZCL) by the Cluster Library Development Board.

3102 **A.1 New Status Enumeration**

3103 A new ZCL status enumeration is required for use with Smart Energy devices, as described in
3104 Table A-1.

3105 **Table A-1– Additional Status Enumeration**

3106 Enumerated Status	Value	Description
NOTIFICATION_PENDING	0x9A	The command has been received and is being processed.

3107 **A.2 New Attribute Reporting Status Indication**

3108 A new *Attribute Reporting Status* attribute is to be added to all Smart Energy clusters.
3109 Consideration will be given to extending this to all ZCL clusters. Details for this attribute are
3110 shown in Table A-2:

3111 **Table A-2– Attribute Reporting Status Attribute**

3112 Identifier	Name	Type	Range	Access	Default	Man./Opt.
0xFFFFE	<i>Attribute Reporting Status</i>	8-bit Enumeration	0x00 to 0xFF	Read Only	-	O

3113 **A.2.1 Attribute Reporting Status Attribute**

3114 When utilizing the *Report Attributes* command, this 8-bit enumeration should be included as the last
3115 attribute in order to indicate that all required attributes have been reported. The enumerated values
3116 for this attribute are outlined in Table A-3:

3117

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

3120 **Annex B ENHANCED INTER-PAN TRANSMISSION MECHANISM**

3121 *NOTE: This annex previously described the requirements for the ‘Anonymous Inter-PAN’*
3122 *mechanism. It has been removed and replaced with a new annex defining the ‘Enhanced’ Inter-*
3123 *PAN mechanism which is to be used in conjunction with devices such as Hand Held Terminals*
3124 *(HHTs).*

3125 *The ‘Enhanced’ Inter-PAN mechanism in this revision of this specification is visionary and is*
3126 *not certifiable. This mechanism may change before reaching certifiable status in a future*
3127 *revision of this specification.*

3128 **B.1 Scope and Purpose**

3129 This annex defines a mechanism whereby ZigBee devices can perform limited exchanges of
3130 information with devices in their local neighborhood without having to form or join the same
3131 ZigBee network. The Enhanced Inter-PAN mechanism is intended to be used in conjunction with
3132 devices such as Hand Held Terminals (HHTs).

3133 The intended destination for the mechanism described here is not the ZigBee specification
3134 [B7], but the relevant application profile documents for applications that make use of the
3135 feature – in particular, the Smart Energy Standard.

3136 **B.2 General Description**

3137 **B.2.1 What Enhanced Inter-PAN Transmission Does**

3138 A schematic view of the how inter-PAN transmission in a ZigBee context works is shown in
3139 Figure B-1.

3140

3141

3142

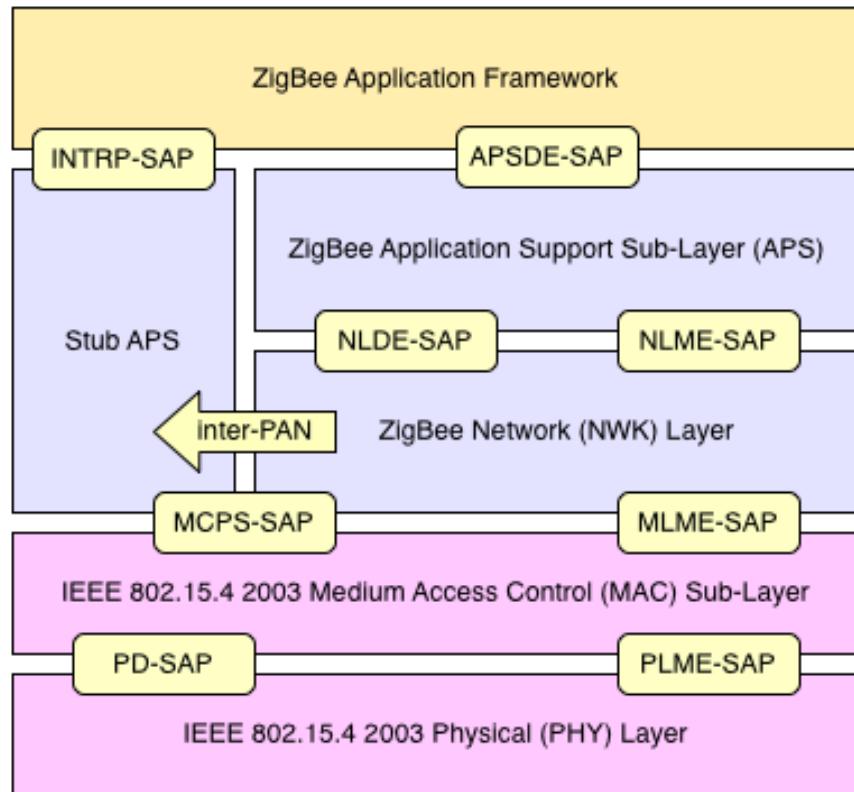


Figure B-1– ZigBee Stack with Stub APS

3143
3144
3145

3146 Inter-PAN data exchanges are handled by a special “stub” of the Application Support Sub-
3147 Layer, which is accessible through a special Service Access Point (SAP), the INTRP-SAP,
3148 parallel to the normal APSDE-SAP. The stub APS performs just enough processing to pass
3149 application data frames to the MAC for transmission and also to pass Inter-PAN application
3150 frames from the MAC to the application on receipt.

3151 The Inter-PAN data exchange architecture does not support simultaneous execution by multiple
3152 application entities. Within a device, only one application entity shall use the Inter-PAN
3153 communications mechanisms.

3154 The Enhanced Inter-PAN specification includes options for its operation and security. Each
3155 profile and feature that uses Inter-PAN will specify the options allowed.

3156 **B.3 Service Specification**

3157 The INTRP-SAP is a data service comprising three primitives.

- 3158 • INTRP-DATA.request - Provides a mechanism for a sending device to request transmission
3159 of an Inter-PAN message.
- 3160 • INTRP-DATA.confirm - Provides a mechanism for a sending device to understand the status
3161 of a previous request to send an Inter-PAN message.
- 3162 • INTRP-DATA.indication - Provides a mechanism for identifying and conveying an Inter-
3163 PAN message received from a sending device.

3164 **B.3.1 The INTRP-DATA.request Primitive**

3165 The INTRP-DATA.request primitive allows an application entity to request data transmission
3166 via the stub APS.

3167 **B.3.1.1 Semantics of the Service Primitive**

3168 The primitive interface is as follows:

3170 INTRP-DATA.request {
3171 SrcAddrMode
3172 DstAddrMode
3173 DstPANId
3174 DstAddress
3175 ProfileId
3176 ClusterId
3177 ASDULength
3178 ASDU
3179 ASDUHandle
3180 TxOptions
3181 }
3182

3183 Parameters of the primitive appear in Table B-1.
3184

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.
ASDCHandle	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)

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	INTRP-DATA.confirm { ASDUHandle Status }
3205	
3206	
3207	

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

3211 **B.3.2.2 When Generated**

3212 This primitive is generated by the stub APS on a ZigBee device and passed to the application in
3213 response to the receipt of a MCPS-DATA.confirm primitive that is a confirmation of a previous
3214 MCPS-DATA.request issued by the stub APS.

3215 **B.3.2.3 Effect on Receipt**

3216 As a result of the receipt of this primitive, the application is informed of the results of an
3217 attempt to send a frame via the stub APS.

3218 **B.3.3 The INTRP-DATA.indication Primitive**

3219 The INTRP-DATA.indication primitive allows the stub APS to inform the next higher layer
3220 that it has received a frame that was transmitted via the stub APS on another device.

3221 **B.3.3.1 Semantics of the Service Primitive**

3222 The primitive interface is as follows:

```
3223
3224    INTRP-DATA.indication                 {
3225                                                 SrcAddrMode
3226                                                 SrcPANId
3227                                                 SrcAddress
3228                                                 DstAddrMode
3229                                                 DstPANId
3230                                                 DstAddress
3231                                                 ProfileId
3232                                                 ClusterId
3233                                                 ASDULength
3234                                                 ASDU
3235                                                 Status
3236                                                 SecurityStatus
3237                                                 LinkQuality
3238                                                 }
```

3239

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 – <i>(aMaxMACFrameSize - 9)</i>	The number of octets in the ASDU to be transmitted.
ASDU	Set of octets	-	The set of octets forming the ASDU to be transmitted.

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**

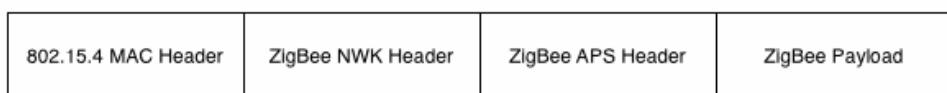
3250 Upon receipt of this primitive the application is informed of the receipt of an application
 3251 frame transmitted, via the stub APS, by a peer device and intended for the receiving device.

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].

3263 Since most of the information contained in the NWK and APS headers is not relevant for
 3264 Inter-PAN transmission, the Inter-PAN frame, shown Figure B-3, contains only a stub of the
 3265 NWK header and the APS header, which provide the information required by the stub APS
 3266 shown in Figure B-4 to do its job.

802.15.4 MAC Header	ZigBee NWK Header	ZigBee APS Header	ZigBee Payload
---------------------	-------------------	-------------------	----------------

Figure B-3– Enhanced Inter-PAN ZigBee Frame

3267

3268

3269

3270

3271

3272

3273

3274

3275

3276

3277

3278

3279

3280

3281

3282

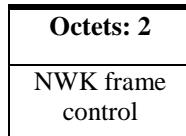


Figure B-4– Stub NWK Header Format

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

3275

3276

3277

3278

3279

3280

3281

3282

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

3283 The stub APS header contains as many as seven fields totaling a maximum of 25 octets in
 3284 length.

3285 The APS frame control field shall be 1 octet in length and is identical in format to the frame
 3286 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

3287 The fields of the frame control field have the following values:

- 3289 • The frame type sub-field shall have a value of 0b11, which is a reserved frame type with
 3290 respect to the [B3].
- 3291 • The delivery mode sub-field may have a value of 0b00, indicating unicast, 0b10, indicating
 3292 broadcast or 0b11 indicating group addressing.
- 3293 • The Security sub-field may have the value 0 for no security or 1 for a secured frame. When
 3294 the Security sub-field is 1 an Auxiliary Header shall be present.
- 3295 • The ACK request sub-field shall have a value of 0 when no ACK is requested and value 1
 3296 when an ACK is requested. When an ACK is requested, the APS Counter field shall be present.
- 3297 • The extended header present sub-field shall have a value of 0, indicating no extended header,
 3298 or a value of 1, indicating that a message has been fragmented and that there is an extended
 3299 header.

3300 The optional Group Address shall be present if and only if the Delivery Mode field has a value
 3301 of 0x0b11 and the delivery mode is 0b11 indicating a group address. If present, it shall contain
 3302 the 16-bit identifier of the group to which the frame is addressed.

3303 The Cluster Identifier field is 2 octets in length and specifies the identifier of the cluster to
 3304 which the frame relates and which shall be made available for filtering and interpretation of
 3305 messages at each device that takes delivery of the frame.

3306 The Profile Identifier is two octets in length and specifies the ZigBee profile identifier for
 3307 which the frame is intended and shall be used during the filtering of messages at each device that
 3308 takes delivery of the frame. When the Profile Identifier is set to that of the Smart Energy profile,
 3309 the Security sub-field of the APS Control field shall be set as follows. If the cluster ID is set to
 3310 0x0019 (Key Establishment), the Security sub-field shall be set to 0; otherwise it shall be set to 1.

3311 **B.5 Frame Processing**

3312 Assuming the INTRP-SAP described above, frames transmitted using the stub APS are
3313 processed as described here.

3314 **B.5.1 Enhanced Inter-PAN Transmission**

3315 On receipt of the INTRP-DATA.request primitive, the stub APS shall construct a stub APS
3316 frame. The header of the stub APS frame shall contain a NWK and an APS frame control field
3317 as described in clause B.4, a cluster identifier field equal to the value of the ClusterId parameter
3318 of the INTRP-DATA.request and a profile identifier field equal to the value of the ProfileId
3319 parameter. If the DstAddrMode parameter of the INTRP-DATA.request has a value of 0x01,
3320 indicating group addressing, then the APS header shall also contain a group address field with
3321 a value corresponding to the value of the DstAddress parameter. The payload of the stub APS
3322 frame shall contain the data payload to be transmitted.

3323 The stub APS frame will then be transmitted using the MCPS-DATA.request primitive of the
3324 MAC sub-layer with key primitive parameters set as follows:

- 3325 • The value of the SrcAddrMode parameter of the MCPS-DATA.request shall always be set to
3326 a value of three, indicating the use of the 64-bit extended address.
- 3327 • The SrcPANId parameter shall be equal to the value of the *macPANID* attribute of the MAC
3328 PIB.
- 3329 • The SrcAddr parameter shall always be equal to the value of the MAC sub- layer constant
3330 *aExtendedAddress*.
- 3331 • If the DstAddrMode parameter of the INTRP-DATA.request primitive has a value of 0x01,
3332 then the DstAddrMode parameter of the MCPS-DATA.request shall have a value of 0x02.
3333 Otherwise, the DstAddrMode parameter of the MCPS-DATA.request shall reflect the value
3334 of the DstAddrMode parameter of the INTRP-DATA.request.
- 3335 • The DstPANId parameter shall have the value given by the DstPANID parameter of the
3336 INTRP-DATA.request primitive.
- 3337 • If the DstAddrMode parameter of the INTRP-DATA.request has a value of 0x01, indicating
3338 group addressing, then the value of the DstAddr parameter of the MCPS-DATA.request shall
3339 be the broadcast address 0xffff. Otherwise, the value of the DstAddr parameter shall reflect
3340 the value of the DstAddress parameter of the INTRP-DATA.request primitive.
- 3341 • The MsduLength parameter shall be the length, in octets, of the stub APS frame.
- 3342 • The Msdu parameter shall be the stub APS frame itself.

- 3343 • If the transmission is a unicast packet, then the value of the TxOptions parameter shall be
3344 0x01, indicating a request for acknowledgement. Otherwise, the TxOptions parameter shall
3345 have a value of 0x00, indicating no options.

3346 On receipt of the MCPS-DATA.confirm primitive from the MAC sub-layer, the stub APS will
3347 invoke the transmit confirmation function with a status reflecting the status returned by the
3348 MAC.

3349 **B.5.2 Enhanced Inter-PAN Reception**

3350 On receipt of the MCPS-DATA.indication primitive from the MAC sub-layer, the receiving
3351 entity - in the case of a ZigBee device this is normally the NWK layer - shall determine
3352 whether the frame should be passed to the stub APS or processed as specified in [B3]. For a
3353 frame that is to be processed by the stub APS, the non-varying sub-fields of both the NWK
3354 frame control field and the APS frame control field must be set exactly as described above.

3355 If the delivery mode sub-field of the APS frame control field of the stub APS header has a
3356 value of 0b11, indicating group addressing, then, if the device implements group
3357 addressing, the value of the group address field shall be checked against the NWK layer
3358 group table, and, if the received value is not present in the table, the frame shall be discarded
3359 with no further processing or action.

3360 On receipt of a frame for processing, the stub APS shall generate an INTRP-DATA.indication
3361 with parameter values as follows:

- 3362 • The value of the SrcAddrMode parameter of the INTRP-DATA.indication shall always be
3363 set to a value of three, indicating the use of the 64-bit extended address
- 3364 • The value of the SrcPANId parameter shall reflect that of the SrcPANId parameter of the
3365 MCPS-DATA.indication.
- 3366 • The SrcAddress parameter of the INTRP-DATA.indication shall always reflect the value of a
3367 64-bit extended address.
- 3368 • Values for the DstAddrMode parameter shall be one of the following:
 - 3369 ○ 0x03, if the DstAddrMode parameter of the INTRP-DATA.indication has a value of
3370 0x03.
 - 3371 ○ 0x02, if the DstAddrMode parameter of the INTRP-DATA.indication has a value of
3372 0x02
- 3373 • The value of the DstPANId parameter of the INTRP-DATA.indication shall reflect the value
3374 of the DstPANId parameter of the MCPS-DATA.indication.

- 3375 • If the DstAddrMode parameter of the INTRP-DATA.indication has a value of 0x01,
3376 indicating group addressing, then the DstAddress parameter of the INTRP-DATA.indication
3377 shall reflect the value of the Group Address field of the stub APS header. Otherwise, the
3378 value of the DstAddress parameter of the INTRP-DATA.indication shall reflect the value of
3379 the DstAddr parameter of the MCPS-DATA.indication.
- 3380 • The value of the ProfileId parameter shall be the same as the value of the Profile Identifier
3381 field of the stub APS header.
- 3382 • The value of the ClusterId parameter shall be the same as the value of the Cluster Identifier
3383 field of the stub APS header.
- 3384 • The ASDULength field shall contain the number of octets in the stub APS frame payload.
- 3385 • The ASDU shall be the stub APS payload itself.
- 3386 • The value of the LinkQuality parameter shall reflect the value of the mpduLinkQuality
3387 parameter of the MCPS-DATA.indication.

3388 **B.6 Initiating an Enhanced Inter-PAN Interaction**

- 3389 1. Unless Inter-PAN is being used in conjunction with Network Joining, it is assumed that there
3390 will be no need to permit joining on the network on which the required receiving device is
3391 present.
- 3392 2. To find the required receiving device with which to interact, the initiating device shall send
3393 Beacon Request packets on each channel, dwelling on each channel as specified by the
3394 ZigBee PRO specification beacon response window.
- 3395 3. The device shall survey all required channels and build a list of devices which have emitted
3396 a beacon that has been heard. If the initiating device has a user interface, the available
3397 devices should be displayed in order that the required receiving device can be chosen. If no
3398 user interface is available, then other method(s) shall be employed to choose a required
3399 receiving device (e.g. strongest signal).
- 3400 4. The device initiates a CBKE process with the required receiving device by sending an
3401 *Initiate Key Establishment Request* command within an unsecured Enhanced Inter-PAN
3402 frame (Security sub-field set to 0).
- 3403 5. The required receiving device may employ techniques to ensure that the initiating device is
3404 acceptable for such communication (e.g. by checking its EUI-64 address against a list), and
3405 refusing the transaction if it is not suitable.

- 3406 6. If the initiating device is acceptable to the required receiving device, the Enhanced Inter-
3407 PAN CBKE process will be allowed to complete, and the resultant shared APS link key shall
3408 be used to secure all further Enhanced Inter-PAN frames (Security sub-field set to 1).

3409 **B.7 Best Practices**

3410 Network Channel Manager Inter-PAN support is not specified in Annex E of the core stack
3411 specification ([B3]). New channel notifications will not be broadcast using Inter-PAN. Inter-
3412 PAN devices which do not receive the network channel change will need to perform the network
3413 discovery procedure described in B.3.4.

3414 **B.8 Security Requirements**

3415 Smart Energy profile devices supporting Enhanced Inter-PAN shall use APS security for all Inter-
3416 PAN frames except for those key establishment messages that do not require it.

3417 In addition, devices shall verify the correct format of all SE messages received over Enhanced
3418 Inter-PAN. Any received message that does not conform to the format described in this
3419 document shall be dropped.

3421 **Annex C KEY ESTABLISHMENT CLUSTER**

3422
3423 The candidate material in this annex, when approved, will be merged into the Foundation
3424 document of the ZigBee Cluster Library (ZCL) by the Cluster Library Development Board.

3425 **C.1 Scope and Purpose**

3426 This Annex specifies a cluster, which contains commands and attributes necessary for managing
3427 secure communication between ZigBee devices.

3428 This Annex should be used in conjunction with the ZigBee Cluster Library, Foundation
3429 Specification (see [B1]), which gives an overview of the library and specifies the frame formats
3430 and general commands used therein.

3431 This version is specifically for inclusion in the Smart Energy profile. The document
3432 which originates from [B4] will continue to be developed in a backward-compatible
3433 manner as a more general secure communication cluster for ZigBee applications as a whole.

3434 **C.2 General Description**

3435 **C.2.1 Introduction**

3436 As previously stated, this document describes a cluster for managing secure communication
3437 in ZigBee. The cluster is for Key Establishment.

3438 **C.2.2 Security Credentials**

3439 Key Establishment requires that the device utilize pre-installed security credentials that are
3440 unique to the device. Depending on the number of cryptographic suites that the device supports,
3441 there may be multiple credentials installed. It is assumed that the device is capable of managing
3442 this and to provide the corresponding credentials based on what suite is being actively used. The
3443 mechanism for negotiating the Key Establishment suite is described in section C.3.1.1.1.

3444 **C.2.3 Network Security**

3445 The Key Establishment Cluster has been designed to be used where the underlying
3446 network security cannot be trusted. As such, no information that is confidential information
3447 will be transported.

3448 **C.2.4 Key Establishment**

3449 To allow integrity and confidentiality of data passed between devices, cryptographic schemes
3450 need to be deployed. The cryptographic scheme deployed in the ZigBee Specification for frame
3451 integrity and confidentiality is based upon a variant of the AES-CCM described in [B15] called

3452 AES-CCM*. This relies on the existence of secret keying material shared between the involved
3453 devices. There are methods to distribute this secret keying material in a trusted manner. However,
3454 these methods are generally not scalable or communication may be required with a trusted key
3455 allocation party over an insecure medium. This leads to the requirement for automated key
3456 establishment schemes to overcome these problems.

3457 Key establishment schemes can either be effected using either a key agreement scheme or a
3458 key transport scheme. The key establishment scheme described in this document uses a key
3459 agreement scheme, therefore key transport schemes will not be considered further in this
3460 document.

3461 A key agreement scheme is where both parties contribute to the shared secret and therefore the
3462 secret keying material to be established is not sent directly; rather, information is exchanged
3463 between both parties that allows each party to derive the secret keying material. Key agreement
3464 schemes may use either symmetric key or asymmetric key (public key) techniques. The party
3465 that begins a key agreement scheme is called the initiator, and the other party is called the
3466 responder.

3467 Key establishment using key agreement involves an initiator and a responder and four steps:

- 3468 1 Establishment of a trust relationship
- 3469 2 Exchange of ephemeral data
- 3470 3 Use of this ephemeral data to derive secret keying material using key agreement
- 3471 4 Confirmation of the secret keying material.

3472 There are two basic types of key establishment which can be implemented:

- 3473 • Symmetric Key Key Establishment
- 3474 • Public Key Key Establishment

3475 **C.2.5 Symmetric Key Key Establishment**

3476 Symmetric Key Key Establishment (SKKE) is based upon establishing a link key based on a
3477 shared secret (master key). If the knowledge of the shared secret is compromised, the
3478 established link key can also be compromised. If the master key is publicly known or is set to a
3479 default value, it is known as Unprotected Key Establishment (UKE). SKKE is the key
3480 establishment method used in the ZigBee specification therefore it will not be considered any
3481 further.

3482 **C.2.6 Public Key Key Establishment**

3483 Public Key Key Establishment (PKKE) is based upon establishing a link key based on shared
3484 static and ephemeral public keys. As the public keys do not require any secrecy, the
3485 established link key cannot be compromised by knowledge of them.

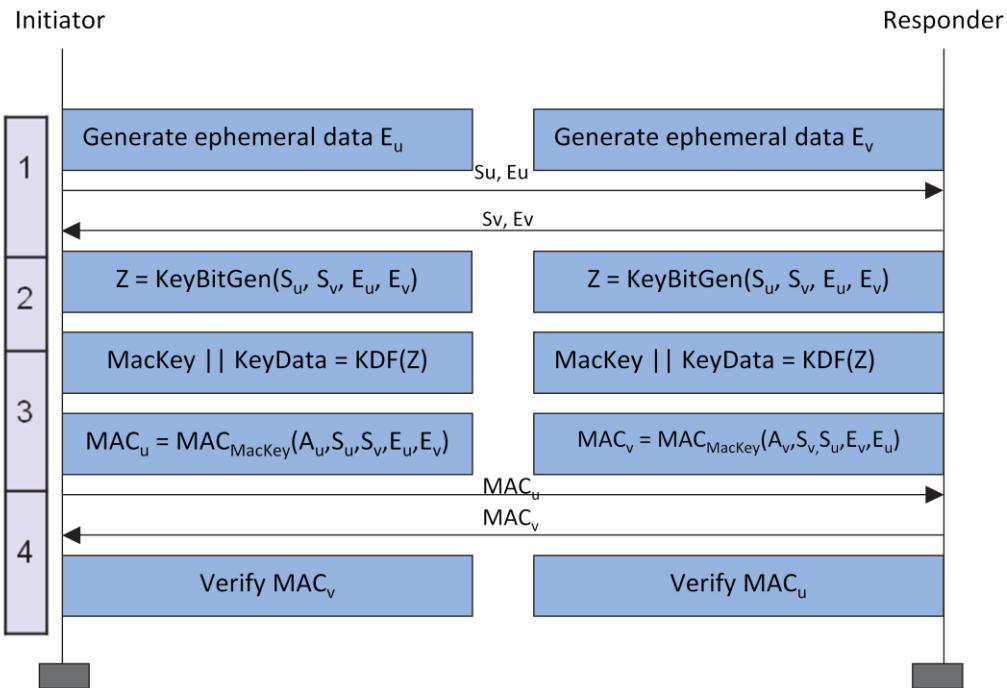
3486 As a device's static public key is used as part of the link key creation, it can either be transported
3487 independently to the device's identity where binding between the two is assumed, or it can be
3488 transported as part of an implicit certificate signed by a Certificate Authority, which provides
3489 authentication of the binding between the device's identity and its public key as part of the key
3490 establishment process. This is called Certificate-Based Key Establishment (CBKE) and is
3491 discussed in more detail in sub-clause C.4.2.

3492 CBKE provides the most comprehensive form of Key Establishment and therefore will be the
3493 method specified in this cluster.

3494 The purpose of the key agreement scheme as described in this document is to produce shared
3495 secret keying material which can be subsequently used by devices using AES-CCM* the
3496 cryptographic scheme deployed in the ZigBee Specification or for any proprietary
3497 security mechanism implemented by the application.

3498 **C.2.7 General Exchange**

3499 The following diagram shows an overview of the general exchange which takes place between
3500 initiator and responder to perform key establishment.



3501

3502

Figure C-1– Overview of General Exchange

3503 The functions are as follows:

- 3504 1 Exchange Static and Ephemeral Data
- 3505 2 Generate Key Bitstream
- 3506 3 Derive MAC key and Key Data
- 3507 4 Confirm Key using MAC

3508 The functions shown in the diagram (Figure C-1) depend on the Key Establishment mechanism.

C.2.7.1 Exchange Static and Ephemeral Data

3509 Figure C-1 shows static data S_U and S_V . For PKKE schemes, this represents a combination of
 3510 the 64-bit device address [B11] and the device's static public key. The identities are needed by
 3511 the MAC scheme and the static public keys are needed by the key agreement scheme.
 3512

3513 Figure C-1 also shows ephemeral data E_U and E_V . For PKKE schemes, this represents the
 3514 public key of a randomly generated key pair.

3515 The static and ephemeral data S_U and E_U are sent to V and the static and ephemeral data S_V and
3516 E_V and are sent to U .

3517 **C.2.7.2 Generate Key Bitstream**

3518 Figure C-1 shows the KeyBitGen function for generating the key bitstream. The function's four
3519 parameters are the identifiers and the ephemeral data for both devices. This ensures the same
3520 key is generated at both ends.

3521 For PKKE schemes, this is the ECMQV key agreement schemes specified in Section 6.2 of
3522 SEC1 [B18]. The static data S_U represents the static public key $Q_{1,U}$ of party U , the static data
3523 S_V represents the static public key $Q_{1,V}$ of party V , the ephemeral data E_U represents the
3524 ephemeral public key $Q_{2,U}$ of party U and the ephemeral data E_V represents the ephemeral
3525 public key $Q_{2,V}$ of party V .

3526 **C.2.7.3 Derive MAC Key and Key Data**

3527 Figure C-1 shows the KDF (KeyDerivation Function) for generating the MAC Key and key
3528 data. The MAC Key is used with a keyed hash message authentication function to generate a
3529 MAC and the key data is the shared secret, e.g. the link key itself required for frame protection.

3530 For PKKE schemes, this is the key derivation function as specified in Section 3.6.1 of SEC1
3531 [B18]. Note there is no *SharedInfo* parameter of the referenced KDF, i.e. it is a null octet string
3532 of length 0.

3533
3534 Figure C-1 also shows generation of the MAC using the MAC Key derived using the KDF using
3535 a message comprised of both static data S_U and S_V and ephemeral data E_U and E_V plus an
3536 additional component A which is different for initiator and responder.

3537 For PKKE schemes, this is the MAC scheme specified in section 3.7 of SEC1 [B18]. The
3538 MAC in the reference is the keyed hash function for message authentication specified
3539 in sub-clause C.4.2.2.6 and the message M is a concatenation of the identity (the 64-bit
3540 device address [B11]) of U , the identity of V and point-compressed octet-string representations of
3541 the ephemeral public keys of parties U and V . The order of concatenation depends on whether
3542 it is the initiator or responder. The additional component A is the single octet 02_{16} for the
3543 initiator and 03_{16} for the responder.

3544 **C.2.7.4 Confirm Key Using MAC**

3545 Figure C-1 shows MACs MAC_U and MAC_V

3546 The MAC MAC_U is sent to V and the MAC MAC_V is sent to U . U and V both calculate the
3547 corresponding MAC and compare it with the data received.

C.3 Cluster List

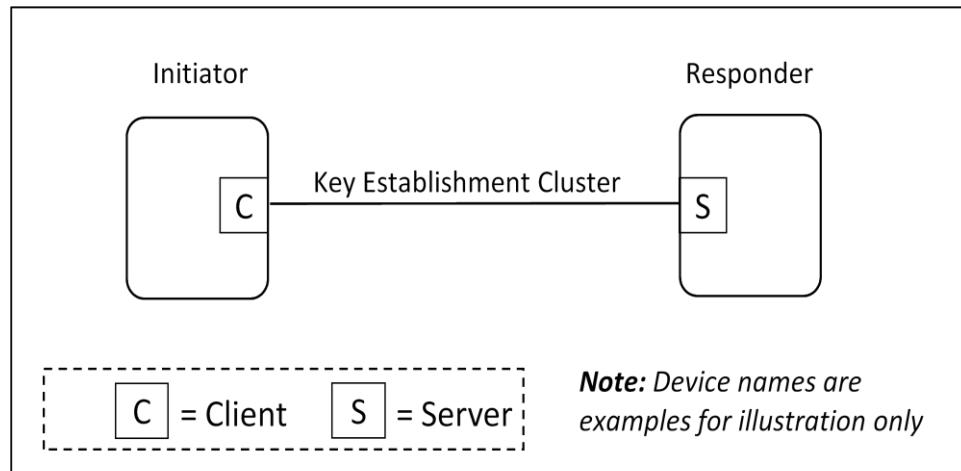
3549 The clusters specified in this document are listed in Table C-1.

3550 For our purposes, any device that implements the client side of this cluster may be considered the
3551 initiator of the secure communication transaction.

3552 **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.

3553



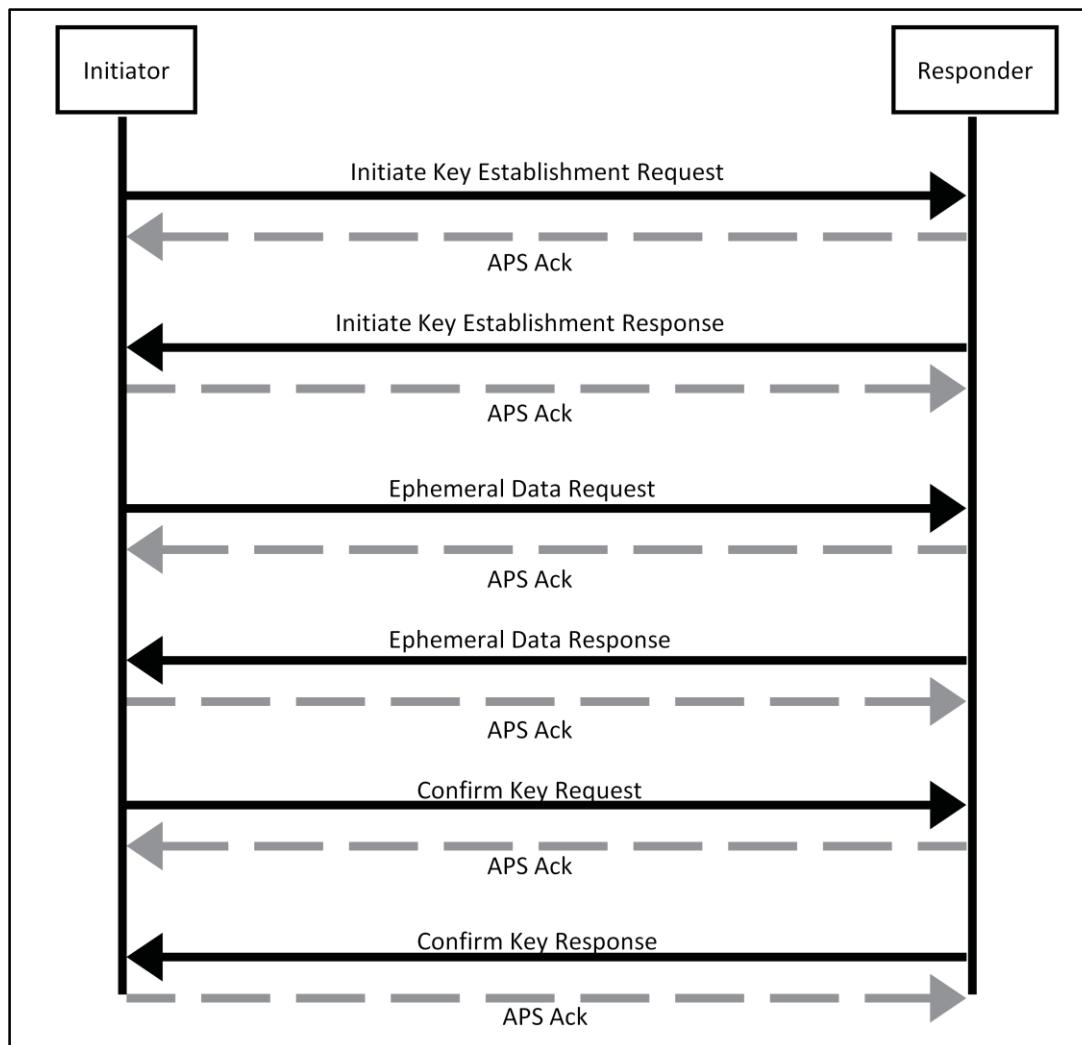
3554

3555 **Figure C-2– Typical Usage of the Key Establishment Cluster**

3556 **C.3.1 Key Establishment Cluster**

3557 **C.3.1.1 Overview**

3558 This cluster provides attributes and commands to perform mutual authentication and establish
3559 keys between two ZigBee devices. Figure C-3 depicts a diagram of a successful key
3560 establishment negotiation.



3561

3562

Figure C-3– Key Establishment Command Exchange

3563 As depicted above, all Key Establishment messages should be sent with APS retries enabled.
 3564 A failure to receive an ACK in a timely manner can be seen as a failure of key establishment.
 3565 No Terminate Key Establishment should be sent to the partner of device that has timed out the
 3566 operation.

3567 The initiator can initiate the key establishment with any active endpoint on the responder
 3568 device that supports the key establishment cluster. The endpoint can be either preconfigured or
 3569 discovered, for example, by using ZDO Match-Desc-req. A link key successfully established
 3570 using key establishment is valid for all endpoints on a particular device. The responder shall
 3571 respond to the initiator using the source endpoint of the initiator's messages as the destination
 3572 endpoint of the responder's messages.

3573 It is expected that the time it takes to perform the various cryptographic computations of
3574 the key establishment cluster may vary greatly based on the device. Therefore rather than set
3575 static timeouts, the *Initiate Key Establishment Request* and *Response* messages will contain
3576 approximate values for how long the device will take to generate the ephemeral data and how
3577 long the device will take to generate confirm key message.

3578 A device performing key establishment can use this information in order to choose a reasonable
3579 timeout for its partner during those operations. The timeout should also take into consideration
3580 the time it takes for a message to traverse the network including APS retries. A minimum
3581 transmission time of 2 seconds is recommended.

3582 For the *Initiate Key Establishment Response* message, it is recommended the initiator wait at
3583 least 2 seconds before timing out the operation. It is not expected that generating an *Initiate Key*
3584 *Establishment Response* will take significant time compared to generating the *Ephemeral Data*
3585 and *Confirm Key* messages.

3586 **C.3.1.1.1 Negotiating the Key Establishment Suite**

3587 Devices may support multiple cryptographic key establishment suites and therefore the client and
3588 server must agree on the suite that is to be used. Devices shall only advertise the suites that they
3589 support and have security credentials for.

3590 The client device is expected to negotiate the key establishment suite with the server, which will
3591 be used for the rest of the key establishment exchange. The initiating device (client) may perform
3592 a Read Attribute request on the *KeyEstablishmentSuite* attribute of the server. It will then
3593 compare its local value of the attribute to the server's value to determine the common set of
3594 suites that are supported by both. The client shall choose the common suite with the highest bit
3595 value and then send the *Initiate Key Establishment Request* message using that suite. If no
3596 common suites are supported, the device shall leave the network.

3597 **C.3.1.2 Server**

3598 **C.3.1.2.1 Dependencies**

3599 The Key Establishment server cluster has no dependencies.

3600 **C.3.1.2.2 Attributes**

3601 For convenience, the attributes defined in this specification are arranged into sets of related
3602 attributes; each set can contain up to 16 attributes. Attribute identifiers are encoded such that the
3603 most significant three nibbles specify the attribute set and the least significant nibble specifies
3604 the attribute within the set. The currently defined attribute sets are listed in Table C-2.

3605 **Table C-2– Key Establishment Attribute Sets**

Attribute Set Identifier	Description
0x000	Information
0x001 – 0xffff	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	<i>KeyEstablishmentSuite</i>	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 schemes for key establishment on the device. A device shall set the corresponding bit to 1 for every cryptographic scheme that it supports. All other cryptographic schemes and reserved bits 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

Command Identifier Field Value	Description	Mandatory/ Optional
0x00	<i>Initiate Key Establishment Request</i>	M
0x01	<i>Ephemeral Data Request</i>	M
0x02	<i>Confirm Key Data Request</i>	M
0x03	<i>Terminate Key Establishment</i>	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

3646 sub-clause C.4.2.2. For *KeyEstablishmentSuite* = 0x0002 ('Crypto Suite 2') the certificate is
3647 specified in sub-clause C.4.2.3.

3648 **C.3.1.2.3.1.2 Effect on Receipt**

3649 If the device does not currently have the resources to respond to a key establishment
3650 request it shall send a *Terminate Key Establishment* command with the result value set to
3651 NO_RESOURCES and the Wait Time field shall be set to an approximation of the time that must
3652 pass before the device will have the resources to process a new Key Establishment Request.

3653 If the receiving device does not support the cryptographic suite specified in the message, it shall
3654 send a *Terminate Key Establishment* message with the status of UNSUPPORTED_SUITE.

3655 If the *KeyEstablishmentSuite* field of the message has more than a single bit selected in the
3656 bitmap, the receiving device shall send a *Terminate Key Establishment* message with the status of
3657 BAD_MESSAGE.

3658 The receiving device shall extract the Issuer field of the implicit certificate received in the
3659 message. It shall then examine all locally installed certificates using the same Cryptographic
3660 suite specified in the received message and compare the Issuer field contained within the
3661 certificate to the issuer within the received certificate. If no locally installed certificates match
3662 the issuer in the received certificate, the device shall send a *Terminate Key Establishment*
3663 command with the result set to UNKNOWN_ISSUER.

3664 If the implicit certificate received in the message is for the 'Crypto Suite 2' Cipher Suite, then
3665 the receiving device shall check the status of the *KeyUsage* field and, if the *Key Agreement*
3666 flag is NOT set, shall send a *Terminate Key Establishment* message with the status of
3667 INVALID_CERTIFICATE. The receiving device shall also check the *Type*, *Curve* and *Hash* fields of
3668 such a certificate, and send a *Terminate Key Establishment* message with the status of
3669 INVALID_CERTIFICATE if any of these fields contains an invalid value.

3670 If the device accepts the request it shall send an *Initiate Key Establishment Response*
3671 command containing its own identity information. It shall set the Key Establishment suite to the
3672 same value as in the received *Initiate Key Establishment Request* message. The identity
3673 information shall correspond to the same suite as specified in the Key Establishment suite. The
3674 device should verify the certificate belongs to the address that the device is communicating with.
3675 The binding between the identity of the communicating device and its address is verifiable using
3676 an out-of-band method.

3677 For all future server messages within the current key establishment negotiation, the Key
3678 Establishment suite value received in this message shall be utilized. If the client receives a
3679 *Terminate Key Establishment* message, or times out the operation, the key establishment suite
3680 value must be renegotiated.

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 (<i>MACU</i>)

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

3706 **Secure Message Authentication Code field:** The Secure Message Authentication Code field
 3707 shall be the octet representation of *MACU* as specified in sub-clause C.4.2.

3708 **C.3.1.2.3.3.2 Effect on Receipt**

3709 If the device is not currently in the middle of negotiating Key Establishment with the sending
 3710 device when it receives this message, it shall send back a Terminate Key Establishment
 3711 message with a result of BAD_MESSAGE. If the device is in the middle of Key Establishment
 3712 with the sender but did not receive this message in response to an *Ephemeral Data Response*
 3713 command, it shall send back a Terminate Key Establishment message with a result of
 3714 BAD_MESSAGE.

3715 On receipt of the *Confirm Key Request* command the responder device shall compare the
 3716 received MACU value with its own reconstructed version of MACU. If the two match the
 3717 responder shall send back MACV by generating an appropriate *Confirm Key Response*
 3718 command. If the two do not match, the responder shall send back a Terminate Key
 3719 Establishment with a result of BAD KEY_CONFIRM and terminate the key establishment.

3720 **C.3.1.2.3.4 Terminate Key Establishment Command**

3721 The *Terminate Key Establishment* command may be sent by either the initiator or responder to
 3722 indicate a failure in the key establishment exchange.

3723 **C.3.1.2.3.4.1 Payload Format**

3724 The *Terminate Key Establishment* command payload shall be formatted as illustrated in Figure
 3725 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

3726 **Figure C-7– Terminate Key Establishment Command Payload**

3727 **Status Field:** The Status field shall be one of the error codes in Table C-6.

3728 **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.

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

3729
 3730 **Wait Time:** This value indicates the minimum amount of time in seconds the initiator device
 3731 should wait before trying to initiate key establishment again. The valid range is 0x00 to 0xFE.

3732 **KeyEstablishmentSuite:** This value will be set the value of the *KeyEstablishmentSuite* attribute.
 3733 It indicates the list of key exchange methods that the device supports.

3734 **C.3.1.2.3.4.2 Effect on Receipt**

3735 On receipt of the *Terminate Key Establishment* command the device shall terminate key
 3736 establishment with the sender. If the device receives a status of BAD_MESSAGE or
 3737 NO_RESOURCES it shall wait at least the time specified in the Wait Time field before trying to
 3738 re-initiate Key Establishment with the device.

3739 If the device receives a status of UNSUPPORTED_SUITE it should examine the
 3740 KeyEstablishmentSuite field to determine if another suite can be used that is supported by
 3741 the partner device. It may re-initiate key establishment using that one of the supported suites
 3742 after waiting the amount of time specified in the Wait Time field. If the device does not
 3743 support any of the types in the KeyEstablishmentSuite field, it should not attempt key
 3744 establishment again with that device.

3745 If the device receives a status of UNKNOWN_ISSUER or BAD_KEY_CONFIRM the device
 3746 should not attempt key establishment again with the device, as it is unlikely that another attempt
 3747 will be successful.

3748 **C.3.1.2.4 Commands Generated**

3749 The server generates the commands detailed in sub-clause C.3.1.3.3, as well as those used for
 3750 reading and writing attributes.

3751 C.3.1.3 Client

3752 C.3.1.3.1 **Dependencies**

3753 The Key Establishment client cluster has no dependencies.

3754 C.3.1.3.2 **Attributes**

3755 For convenience, the attributes defined in this specification are arranged into sets of related
3756 attributes; each set can contain up to 16 attributes. Attribute identifiers are encoded such that the
3757 most significant three nibbles specify the attribute set and the least significant nibble specifies the
3758 attribute within the set. The currently defined attribute sets are listed in Table C-7.

3759

3760 **Table C-7– Key Establishment Attribute Sets**

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

3761

3762 C.3.1.3.2.1 **Information**

3763 The Information attribute set contains the attributes summarized in Table C-8.

3764 **Table C-8– Attributes of the Information Attribute Set**

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

3765

3766

3767

3768 C.3.1.3.2.1.1 ***KeyEstablishmentSuite* Attribute**

3769 The *KeyEstablishmentSuite* attribute is 16-bits in length and specifies ALL the cryptographic
3770 schemes for key establishment on the device. A device shall set the corresponding bit to 1 for
3771 every cryptographic scheme that is supports. All other cryptographic schemes and reserved bits
3772 shall be set to 0. This attribute shall be set to one of the non-reserved values listed in Table C-9.

3773 Although, for backwards compatibility, the Type cannot be changed, this 16-bit Enumeration
3774 should be treated as if it were a 16-bit BitMap.

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	<i>Initiate Key Establishment Response</i>	M
0x01	<i>Ephemeral Data Response</i>	M
0x02	<i>Confirm Key Data Response</i>	M
0x03	<i>Terminate Key Establishment</i>	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)

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

3790 **Requested Key Establishment Suite:** This will be the type of *KeyEstablishmentSuite* that the
 3791 initiator has requested be used for the key establishment exchange. The responder device shall set
 3792 a single bit in the bitmask indicating that it has accepted the requested suite; all other bits shall be
 3793 set to zero.

3794 **Ephemeral Data Generate Time:** This value indicates approximately how long in seconds the
 3795 responder device takes to generate the Ephemeral Data Response message. The valid range is
 3796 0x00 to 0xFE.

3797 **Confirm Key Generate Time:** This value indicates approximately how long the responder
 3798 device will take in seconds to generate the Confirm Key Response message. The valid range
 3799 is 0x00 to 0xFE.

3800 **Identity field:** The *Identity* field shall be the block of octets containing the implicit certificate
 3801 CERTU. For *KeyEstablishmentSuite* = 0x0001 ('Crypto Suite 1'), the certificate is specified in
 3802 sub-clause C.4.2.2. For *KeyEstablishmentSuite* = 0x0002 ('Crypto Suite 2'), the certificate is
 3803 specified in sub-clause C.4.2.2.3.

3804 C.3.1.3.3.1.2 Effect on Receipt

3805 If the device is not currently in the middle of negotiating Key Establishment with the sending
 3806 device when it receives this message, it shall send back a Terminate Key Establishment
 3807 message with a result of BAD_MESSAGE. If the device is in the middle of Key Establishment
 3808 with the sender but did not receive this message in response to an *Initiate Key Establishment*
 3809 Request command, it shall send back a Terminate Key Establishment message with a result of
 3810 BAD_MESSAGE.

3811 If the receiving device does not support the key establishment suite specified in the message, it
 3812 shall send a *Terminate Key Establishment* message with the status of UNSUPPORTED_SUITE.

3813 If the *Requested Key Establishment Suite* field of the message has more than a single bit selected
 3814 in the bitmap, the receiving device shall send a *Terminate Key Establishment* message with the
 3815 status of BAD_MESSAGE.

3816 On receipt of this command the device shall check the Issuer field of the device's implicit
 3817 certificate. If the Issuer field does not contain a value that corresponds to a known Certificate
 3818 Authority, the device shall send a *Terminate Key Establishment* command with the status
 3819 value set to UNKNOWN_ISSUER. If the device does not currently have the resources to
 3820 respond to a key establishment request it shall send a *Terminate Key Establishment* command
 3821 with the status value set to NO_RESOURCES and the Wait Time field shall be set to an
 3822

3824 approximation of the time that must pass before the device has the resources to process the
 3825 request.

3826 The receiver shall verify that the *KeyEstablishmentSuite* in the *Initiate Key Establishment*
 3827 *Response* matches the value that was sent in the *Initiate Key Establishment Request*. If the values
 3828 do not match then the device shall send a *Terminate Key Establishment Request* with
 3829 **UNSUPPORTED_SUITE**.

3830 If the implicit certificate received in the message is for the ‘Crypto Suite 2’ Cipher Suite, then
 3831 the receiving device shall check the status of the *KeyUsage* field and, if the *Key Agreement*
 3832 flag is NOT set, shall send a *Terminate Key Establishment* message with the status of
 3833 **INVALID_CERTIFICATE**. The receiving device shall also check the *Type*, *Curve* and *Hash* fields of
 3834 such a certificate, and send a *Terminate Key Establishment* message with the status of
 3835 **INVALID_CERTIFICATE** if any of these fields contains an invalid value.

3836 If the device accepts the response it shall send an *Ephemeral Data Request* command. The device
 3837 should verify the certificate belongs to the address that the device is communicating with. The
 3838 binding between the identity of the communicating device and its address is verifiable using out-
 3839 of-band method.

3840 For all future client messages within the current key establishment negotiation, the Key
 3841 Establishment suite value received in this message shall be utilized. If the client receives a
 3842 *Terminate Key Establishment* message, or times out the operation, the key establishment suite
 3843 value must be renegotiated.

3844 **C.3.1.3.3.2 Ephemeral Data Response Command**

3845 The *Ephemeral Data Response* command allows a device to communicate its ephemeral data
 3846 to another device that previously requested it.

3847 **C.3.1.3.3.2.1 *Payload Format***

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

3848 **Figure C-9–Ephemeral Data Response Command Payload**

3849 **C.3.1.3.3.2.2 *Effect on Receipt***

3850 If the device is not currently in the middle of negotiating Key Establishment with the sending device
 3851 when it receives this message, it shall send back a *Terminate Key Establishment* message with a
 3852 result of **BAD_MESSAGE**. If the device is in the middle of Key Establishment with the sender but did
 3853 not receive this message in response to an *Ephemeral Data Request* command, it shall send back a
 3854 *Terminate Key Establishment* message with a result of **BAD_MESSAGE**.

3855 The length of the frame shall correlate to the current key establishment suite that has been negotiated by
 3856 the client and server (refer to Table C-14 for relevant sizes). If the length of the Ephemeral Data is
 3857 shorter than the expected length according to the cryptographic suite, the responder shall send back a
 3858 Terminate Key Establishment message with a result of BAD_MESSAGE.

3859 On receipt of this command if the device can handle the request it shall perform key generation, key
 3860 derivation, and MAC generation. If successful it shall generate an appropriate *Confirm Key*
 3861 *Request* command, otherwise it shall generate a Terminate Key Establishment with a result value
 3862 of NO_RESOURCES.

3863 **C.3.1.3.3.3 Confirm Key Response Command**

3864 The *Confirm Key Response* command allows the responder to verify the initiator has derived the
 3865 same secret key. This is done by sending the initiator a cryptographic hash generated using the
 3866 keying material and the identities and ephemeral data of both parties.

3867 **C.3.1.3.3.3.1 *Payload Format***

3868 The *Confirm Key Response* command payload shall be formatted as illustrated in Figure C-10.

3869

Octets	16
Data Type	Octets (non-ZCL Data Type)
Field Name	Secure Message Authentication Code (<i>MACV</i>)

3870 **Figure C-10– Confirm Key Response Command Payload**

3871 **Secure Message Authentication Code field:** The Secure Message Authentication Code field
 3872 shall be the octet representation of *MACV* as specified in sub-clause C.4.2.

3873 **C.3.1.3.3.3.2 *Effect on Receipt***

3874 If the device is not currently in the middle of negotiating Key Establishment with the sending
 3875 device when it receives this message, it shall send back a Terminate Key Establishment
 3876 message with a result of BAD_MESSAGE. If the device is in the middle of Key Establishment
 3877 with the sender but did not receive this message in response to an *Confirm Key Request*
 3878 command, it shall send back a Terminate Key Establishment message with a result of
 3879 BAD_MESSAGE.

3880 On receipt of the *Confirm Key Response* command the initiator device shall compare the
 3881 received *MACV* value with its own reconstructed version of the *MACV*. If the two match
 3882 then the initiator can consider the key establishment process to be successful. If the two do

3883 not match, the initiator should send a *Terminate Key Establishment* command with a result of
 3884 BAD_KEY_CONFIRM.

3885 C.3.1.3.3.4 Terminate Key Establishment Command

3886 The *Terminate Key Establishment* command may be sent by either the initiator or responder to
 3887 indicate a failure in the key establishment exchange.

3888 **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

3889 **Figure C-11– Terminate Key Establishment Command Payload**

3890

3891 **Status field:** The Status field shall be one of the following error codes.

3892

3893

3894

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

3895

3896 **Wait Time:** This value indicates the minimum amount of time in seconds the initiator device
3897 should wait before trying to initiate key establishment again. The valid range is 0x00 to 0xFE.

3898 **KeyEstablishmentSuite:** This value will be set the value of the *KeyEstablishmentSuite* attribute.
3899 It indicates the list of key exchange methods that the device supports.

3900 **C.3.1.3.3.4.2 Effect on Receipt**

3901 On receipt of the *Terminate Key Establishment* command the device shall terminate key
3902 establishment with the sender. If the device receives a status of BAD_MESSAGE or
3903 NO_RESOURCES it shall wait at least the time specified in the Wait Time field before trying to
3904 re-initiate Key Establishment with the device.

3905 If the device receives a status of UNKNOWN_SUITE it should examine the
3906 *KeyEstablishmentSuite* field to determine if another suite can be used that is supported by the
3907 partner device. It may re-initiate key establishment using that one of the supported suites after
3908 waiting the amount of time specified in the Wait Time field. If the device does not support any of
3909 the types in the *KeyEstablishmentSuite* field, it should not attempt key establishment again with
3910 that device.

3911 If the device receives a status of UNKNOWN_ISSUER or BAD_KEY_CONFIRM *the device*
3912 *should not attempt key establishment again with the device*, as it is unlikely that another attempt
3913 will be successful.

3914 **C.3.1.3.4 Commands Generated**

3915 The client generates the commands detailed in sub-clause C.3.1.2.3, as well as those used for
3916 reading and writing attributes.

3917 **C.4 Application Implementation**

3918 **C.4.1 Network Security for Smart Energy Networks**

3919 The underlying network security for Smart Energy networks is assumed to be ZigBee
3920 Standard security using pre-configured link keys.

3921 A temporary link key for a joining device is produced by performing the cryptographic
3922 hash function on a random number assigned to the joining device (e.g. serial number) and the
3923 device identifier, which is the device's 64-bit IEEE address [B11].

3924 The joining device's assigned random number is then conveyed to the utility via an out-of-band
3925 mechanism (e.g. telephone call, or web site registration). The utility then commissions the
3926 energy service interface (ESI) at the premises where the joining device is by installing the
3927 temporary link key at the ESI on the back channel.

3928 When the joining device powers up, it will also create a temporary link key as above and
3929 therefore at the time of joining both the joining device and the ESI have the same temporary link
3930 key, which can be used to transport the network key securely to the joining device.

3931 At this point, the device will be considered joined and authenticated as far as network security
3932 is concerned. The secure communication cluster can now be invoked to replace the temporary
3933 link key with a more secure link key based on public key cryptography.

3934 **C.4.2 Certificate-Based Key Establishment**

3935 The Certificate-Based Key-Establishment (CBKE) solution uses public-key technology with
3936 digital certificates and root keys. Each device has a private key and a digital certificate that is
3937 signed by a Certificate Authority (CA).

3938 The digital certificate includes:

- 3939 • Reconstruction data for the device's public key
- 3940 • The device's extended 64-bit IEEE address
- 3941 • Profile specific information (e.g., the device class, network id, object type, validity date, etc.).

3942 Certificates provide a mechanism for cryptographically binding a public key to a device's
3943 identity and characteristics.

3944 Trust for a CBKE solution is established by provisioning a CA root key and a digital
3945 certificate to each device. A CA root key is the public key paired with the CA's private key. A
3946 CA uses its private key to sign digital certificates and the CA root key is used to verify these
3947 signatures. The trustworthiness of a public key is confirmed by verifying the CA's signature of
3948 the digital certificate. Certificates can be issued either by the device manufacturer, the device
3949 distributor, or the end customer. For example, in practical situations, the CA may be a computer
3950 (with appropriate key management software) that is kept physically secure at the end
3951 customer's facility or by a third-party.

3952 At the end of successful completion of the CBKE protocol the following security services are
3953 offered:

- 3954 • Both devices share a secret link key
- 3955 • Implicit Key Authentication: Both devices know with whom they share this link key.
- 3956 • Key Confirmation: Each device knows that the other device actually has computed the key
3957 correctly

- 3958 • No Unilateral Key Control: No device has complete control over the shared link key that is
3959 established.
- 3960 • Perfect Forward Secrecy: if the private key gets compromised none of future and past
3961 communications are exposed
- 3962 • Known Key Security resilience: Each shared link key created per session is unique

3963 **C.4.2.1 Notation and Representation**

3964 **C.4.2.1.1 Strings and String Operations**

3965 A string is a sequence of symbols over a specific set (e.g., the binary alphabet {0,1} or the set
3966 of all octets). The length of a string is the number of symbols it contains (over the same
3967 alphabet). The right-concatenation of two strings x and y of length m and n respectively
3968 (notation: $x // y$), is the string z of length $m+n$ that coincides with x on its leftmost m symbols
3969 and with y on its rightmost n symbols. An octet is a bit string of length 8.

3970 **C.4.2.1.2 Integers and their Representation**

3971 Throughout this specification, the representation of integers as bit strings or octet strings shall be
3972 fixed. All integers shall be represented as binary strings in most-significant-bit first order and
3973 as octet strings in most-significant-octet first order. This representation conforms to the
3974 convention in Section 2.3 of SEC1 [B18].

3975 **C.4.2.1.3 Entities**

3976 Throughout this specification, each entity shall be a DEV and shall be uniquely identified by
3977 its 64-bit IEEE device address [B11]. The parameter *entlen* shall have the integer value 64.

3978 **C.4.2.2 Cryptographic Suite 1 Building Blocks**

3979 The following cryptographic primitives and data elements are defined for use with the CBKE
3980 ‘Crypto Suite 1’ Cipher suite protocol specified in this document.

3981 **C.4.2.2.1 Elliptic-Curve Domain Parameters**

3982 The elliptic curve domain parameters used by this Cryptographic suite shall be those for the
3983 curve ‘sect163k1’ as specified in section 3.4.1 of SEC2 [B19].

3984 All elliptic-curve points (and operations in this section) used by the ‘Crypto Suite 1’ Cipher Suite
3985 shall be (performed) on this curve.

3986 **C.4.2.2.2 Elliptic-Curve Point Representation**

3987 All elliptic-curve points in the Cryptographic Suite 1 shall be represented as point compressed
3988 octet strings as specified in sections 2.3.3 and 2.3.4 of SEC1 [B18]. Thus, each elliptic-curve
3989 point Cryptographic Suite 1 can be represented in 22 bytes.

3990 C.4.2.2.3 **Elliptic-Curve Key Pair**

3991 An elliptic-curve-key pair consists of an integer d and a point Q on the curve determined by
3992 multiplying the generating point G of the curve by this integer (i.e., $Q=dG$) as specified in section
3993 3.2.1 of SEC1 [B18]. Here, Q is called the public key, whereas d is called the private key; the
3994 pair (d, Q) is called the key pair. Each private key shall be represented as specified in section
3995 2.3.7 of SEC1 [B18]. Each public key shall be represented as defined in sub-clause C.4.2.1.2 of
3996 this document.

3997 C.4.2.2.4 **ECC Implicit Certificates**

3998 The exact format of the 48-byte implicit certificate IC_U used with CBKE scheme shall be
3999 specified as follows:

4000
$$IC_U = PublicReconstrKey \parallel Subject \parallel Issuer \parallel ProfileAttributeData$$

4001 Where,

4002 1 *PublicReconstrKey*: the 22-byte representation of the public-key reconstruction data BEU as
4003 specified in the implicit certificate generation protocol, which is an elliptic-curve point as
4004 specified in sub-clause C.4.2.2.2 (see SEC4 [B18]);

4005 2 *Subject*: the 8-byte identifier of the entity U that is bound to the public-key reconstruction
4006 data *BEU* during execution of the implicit certificate generation protocol (i.e., the extended,
4007 64-bit IEEE 802.15.4 address [B11] of the device that purportedly owns the private key
4008 corresponding to the public key that can be reconstructed with *PublicReconstrKey*);

4009 3 *Issuer*: the 8-byte identifier of the CA that creates the implicit certificate during the execution
4010 of the implicit certificate generation protocol (the so-called Certificate Authority).

4011 4 *ProfileAttributeData*: the 10-byte sequence of octets that can be used by a ZigBee profile for
4012 any purpose. The first two bytes of this sequence is reserved as a profile identifier, which
4013 must be defined by another ZigBee standard.

4014 5 The string I_U as specified in Step 6 of the actions of the CA in the implicit certificate
4015 generation protocol (see section SEC4 [B22]) shall be the concatenation of the *Subject*,
4016 *Issuer*, and *ProfileAttributeData*:

4017
$$I_U = Subject \parallel Issuer \parallel ProfileAttributeData$$

4018 C.4.2.2.5 **Block-Cipher**

4019 The block-cipher used in this specification shall be the Advanced Encryption Standard AES-
4020 128, as specified in FIPS Pub 197 [B16]. This block-cipher has a key size that is equal to the
4021 block size, in bits, i.e., *keylen*= 128.

4022 C.4.2.2.6 **Cryptographic Hash Function**

4023 The cryptographic hash function used in this specification shall be the blockcipher based
4024 cryptographic hash function specified in Annex B.6 in [B3], with the following
4025 instantiations:

4026 1 Each entity shall use the block-cipher E as specified in sub-clause B.1.1 in [B3].

4027 2 All integers and octets shall be represented as specified in sub-clause C.4.2.1.

4028 The Matyas-Meyer-Oseas hash function (specified in Annex B.6 in [B3]) has a message digest
4029 size *hashlen* that is equal to the block size, in bits, of the established blockcipher.

4030 C.4.2.2.7 **Keyed Hash Function for Message Authentication**

4031 The keyed hash message authentication code (HMAC) used in this specification shall be
4032 HMAC, as specified in the FIPS Pub 198 [B17] with the following instantiations:

4033 1 Each entity shall use the cryptographic hash *H* function as specified in sub- clause C.4.2.2.6;

4034 2 The block size *B* shall have the integer value 16 (this block size specifies the length of the
4035 data integrity key, in bytes, that is used by the keyed hash function, i.e., it uses a 128-bit data
4036 integrity key). This is also *MacKeyLen*, the length of *MacKey*.

4037 3 The output size *HMAClen* of the HMAC function shall have the same integer value as the
4038 message digest parameter *hashlen* as specified in sub- clause C.4.2.2.6.

4039 C.4.2.2.8 **Derived Shared Secret**

4040 The derived shared secret *KeyData* is the output of the key establishment. *KeyData* shall
4041 have length *KeyDataLen* of 128 bits.

4042 C.4.2.3 Cryptographic Suite 2 Building Blocks

4043 The elliptic curve domain parameters used by this Cipher suite shall be those for the curve
4044 “sect283k1” as specified in section 3.4.1 of SEC2 [B23].

4045 All elliptic-curve points (and operations in this section) used by the ‘Crypto Suite 2’ Cipher Suite
4046 shall be (performed) on this curve.

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 IC_U 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 $IC_U = Type \parallel SerialNo \parallel Curve \parallel Hash \parallel Issuer \parallel ValidFrom \parallel ValidTo \parallel Subject \parallel KeyUsage \parallel PublicReconstrKey$

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*;

- 4077 **7** *ValidTo*: a 4-byte number giving the seconds from the *ValidFrom* time for which the
4078 certificate is considered valid. A number less than 0xFFFFFFFF gives the number in seconds
4079 while 0xFFFFFFFF indicates an infinite number of seconds;
- 4080 **8** *Subject*: the 8-byte identifier of the entity *U* that is bound to the public-key reconstruction
4081 data *BEU* during execution of the implicit certificate generation protocol (i.e., the extended,
4082 64-bit IEEE 802.15.4 address [B11] of the device that purportedly owns the private key
4083 corresponding to the public key that can be reconstructed with *PublicReconstrKey*);
- 4084 **9** *KeyUsage*: 1-byte identifier indicating the key usage. The complete bit string is defined in
4085 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

4087
4088 For usage of the ‘Crypto Suite 2’ Cipher Suite for Key Establishment, bit 3 shall be set;

4089 **10** *PublicReconstrKey*: the 37-byte representation of the public-key reconstruction data BEU as
4090 specified in the implicit certificate generation protocol, which is an elliptic-curve point as
4091 specified in sub-clause C.4.2.2.2 (see SEC4 [B18]).

4092
4093 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 0xD)
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

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;

4120 The following additional information shall have been unambiguously established between
4121 devices operating the implicit certificate scheme:

- 4122 **1** Each entity shall have obtained information regarding the infrastructure that will be used for
4123 the operation of the implicit certificate scheme - including a certificate format and certificate
4124 generation and processing rules (see SEC4 [B18]);
- 4125 **2** Each entity shall have access to an authentic copy of the elliptic-curve public keys of one or
4126 more certificate authorities that act as CA for the implicit certificate scheme (SEC4 [B18]).

4127 The methods by which this information is to be established are outside the scope of this
4128 standard.

4129 The methods used during the CBKE protocol are described below. The parameters used by these
4130 methods are described in Table C-14.

4131 **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.
QE _U	22	37	The ephemeral public key generated by the initiator device (denoted $Q_{2,U}$ in the Elliptic Curve MQV scheme in SEC1 [B18]).
QE _V	22	37	The ephemeral public key generated by the responder device (denoted $Q_{2,V}$ 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 $(0216 // ID_U // ID_V // QE_U // QE_V)$ 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 QE_U and QE_V 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]).

MACV	16	16	The secure message authentication code generated by the responder device (where the message M is $0316 // ID_V // ID_U // QEV // QEU$) 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 $CERT_U$ 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 $CERT_V$ 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 $CERT_V$ (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 Establishment* command.
 4150 The responder device examines $CERT_U$ (formatted as IC_U as described in sub-clause C.4.2.2.4),
 4151 confirms that the *Subject* identifier is the purported owner of the 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:

-
- 4157 **1** The elliptic curve domain parameters shall be as specified in sub- clause C.4.2.2.1;
- 4158 **2** The KDF shall use the cryptographic hash function specified in sub- clause C.4.2.2.2;
- 4159 **3** The static public key $Q_{I,U}$ shall be the static public key of the initiator;
- 4160 **4** The ephemeral public key $Q_{2,U}$ shall be an ephemeral public key of the initiator generated as
4161 part of this transaction;
- 4162 **5** The static public key $Q_{I,V}$ shall be the static public key of the responder obtained from the
4163 responder's certificate communicated to the initiator by the responder;
- 4164 **6** The ephemeral public key $Q_{2,V}$ shall be based on the point-compressed octet string
4165 representation QEV of an ephemeral key of the responder communicated to the initiator by
4166 the responder;
- 4167 **7** The KDF parameter *keydatalen* shall be *MacKeyLen* + *KeyDataLen*, where *MacKeyLen* is
4168 the length of *MacKey* and *KeyDataLen* is the length of *KeyData*;
- 4169 **8** The parameter *SharedInfo* shall be the empty string;
- 4170 The initiator device derives the keying material *MacKey* and *KeyData* from the output *K* as
4171 specified in section 3.6.1 of SEC1 [B18] by using *MacKey* as the leftmost *MacKeyLen* octets
4172 of *K* and *KeyData* as the rightmost *KeyDataLen* octets of *K*. *KeyData* is used subsequently as the
4173 shared secret and *MacKey* is used for key confirmation.

4174 **C.4.2.4.3.2 Responder**

4175 The responder performs the Elliptic Curve MQV scheme as specified in section 6.2 of SEC1
4176 [B18] with the following instantiations:

- 4177 **1** The elliptic curve domain parameters shall be as specified in sub- clause C.4.2.2.1;
- 4178 **2** The KDF shall use the cryptographic hash function specified in sub- clause C.4.2.2.2;
- 4179 **3** The static public key $Q_{I,U}$ shall be the static public key of the initiator obtained from the
4180 initiator's certificate communicated to the responder by the initiator;
- 4181 **4** The ephemeral public key $Q_{2,U}$ shall be based on the point-compressed octet string
4182 representation QEU of an ephemeral key of the initiator communicated to the responder by
4183 the initiator;
- 4184 **5** The static public key $Q_{I,V}$ shall be the static public key of the responder;

- 4185 **6** The ephemeral public key $Q_{2,V}$ shall be an ephemeral public key of the responder generated
4186 as part of this transaction;
- 4187 **7** The KDF parameter *keydatalen* shall be *MacKeyLen + KeyDataLen*, where *MacKeyLen* is
4188 the length of *MacKey* and *KeyDataLen* is the length of *KeyData*;
- 4189 **8** The parameter *SharedInfo* shall be the empty string;
- 4190 The responder device derives the keying material *MacKey* and *KeyData* from the output *K* as
4191 specified in section 3.6.1 of SEC1 [B18] by using *MacKey* as the leftmost *MacKeyLen* octets
4192 of *K* and *KeyData* as the rightmost *KeyDataLen* octets of *K*. *KeyData* is used subsequently as the
4193 shared secret and *MacKey* is used for key confirmation.

4194 **C.4.2.4.4 Confirm Keys**

4195 **C.4.2.4.4.1 Initiator**

4196 The initiator device uses *MacKey* to compute its message authentication code *MACU* and
4197 sends it to the responder device by using the *Confirm Key* command via the Key Establishment
4198 Cluster Client.

4199 The initiator device uses *MacKey* to confirm the authenticity of the responder by calculating
4200 *MACV* and comparing it with that sent by the responder.

4201 **C.4.2.4.4.2 Responder**

4202 The responder device uses *MacKey* to compute its message authentication code *MACV* and
4203 sends it to the initiator device by using the *Confirm Key* response command via the Key
4204 Establishment Cluster Server.

4205 The responder device uses *MacKey* to confirm the authenticity of the initiator by calculating
4206 *MACU* and comparing it with that sent by the initiator.

4207 **C.5 Key Establishment Test Vectors for Cryptographic Suite 1**

4208 The following details the key establishment exchange data transformation and validation of
4209 test vectors for a pair of Smart Energy devices using Certificate based key exchange (CBKE)
4210 using Elliptical Curve Cryptography (ECC).

4211 **C.5.1 Preconfigured Data**

4212 Each device is expected to have been preinstalled with security information prior to initiating
4213 key establishment. The preinstalled data consists of the Certificate Authority's Public Key, a
4214 device specific certificate, and a device specific private key.

4215 **C.5.1.1 CA Public Key**

4216 The following is the Certificate Authority's Public Key.

4217 02 00 FD E8 A7 F3 D1 08
4218 42 24 96 2A 4E 7C 54 E6
4219 9A C3 F0 4D A6 B8

4220 **C.5.1.2 Responder Data**

4221 The following is the certificate for device 1. The device has an IEEE of (>)0000000000000001,
4222 and will be the responder.

4223 03 04 5F DF C8 D8 5F FB
4224 8B 39 93 CB 72 DD CA A5
4225 5F 00 B3 E8 7D 6D 00 00
4226 00 00 00 00 00 01 54 45
4227 53 54 53 45 43 41 01 09
4228 00 06 00 00 00 00 00 00

4229 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 00 01
Issuer	54 45 53 54 53 45 43 41
Attributes	01 09 00 06 00 00 00 00 00 00

4230

4231 The private key for device 1 is as follows:

4232 00 b8 a9 00 fc ad eb ab
4233 bf a3 83 b5 40 fc e9 ed
4234 43 83 95 ea a7

4235 The public key for device 1 is as follows:

4236 03 02 90 a1 f5 c0 8d ad
4237 5f 29 45 e3 35 62 0c 7a
4238 98 fa c4 66 66 a1

4239 **C.5.1.3 Initiator Data**

4240 The following is the certificate for device 2. The device has an IEEE of (>)0000000000000002,
4241 and will be the initiator.

4242 02 06 15 E0 7D 30 EC A2
4243 DA D5 80 02 E6 67 D9 4B

4244 C1 B4 22 39 83 07 00 00
4245 00 00 00 00 00 02 54 45
4246 53 54 53 45 43 41 01 09
4247 00 06 00 00 00 00 00 00

4248 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 00 02
Issuer	54 45 53 54 53 45 43 41
Attributes	01 09 00 06 00 00 00 00 00 00 00 00

4249
4250 The private key for device 2 is as follows:
4251 01 E9 DD B5 58 0C F7 2E
4252 CE 7F 21 5F 0A E5 94 E4
4253 8D F3 E7 FE E8

4254 The public key for device 2 is:
4255 03 02 5B BA 38 D0 C7 B5
4256 43 6B 68 DF 72 8F 09 3E
4257 7A 1D 6C 43 7E 6D

4258 **C.5.2 Key Establishment Messages**

4259 The following is the basic flow of messages back and forth between the initiator and the
4260 responder performing key establishment using the Key Establishment Cluster.

4261

4262

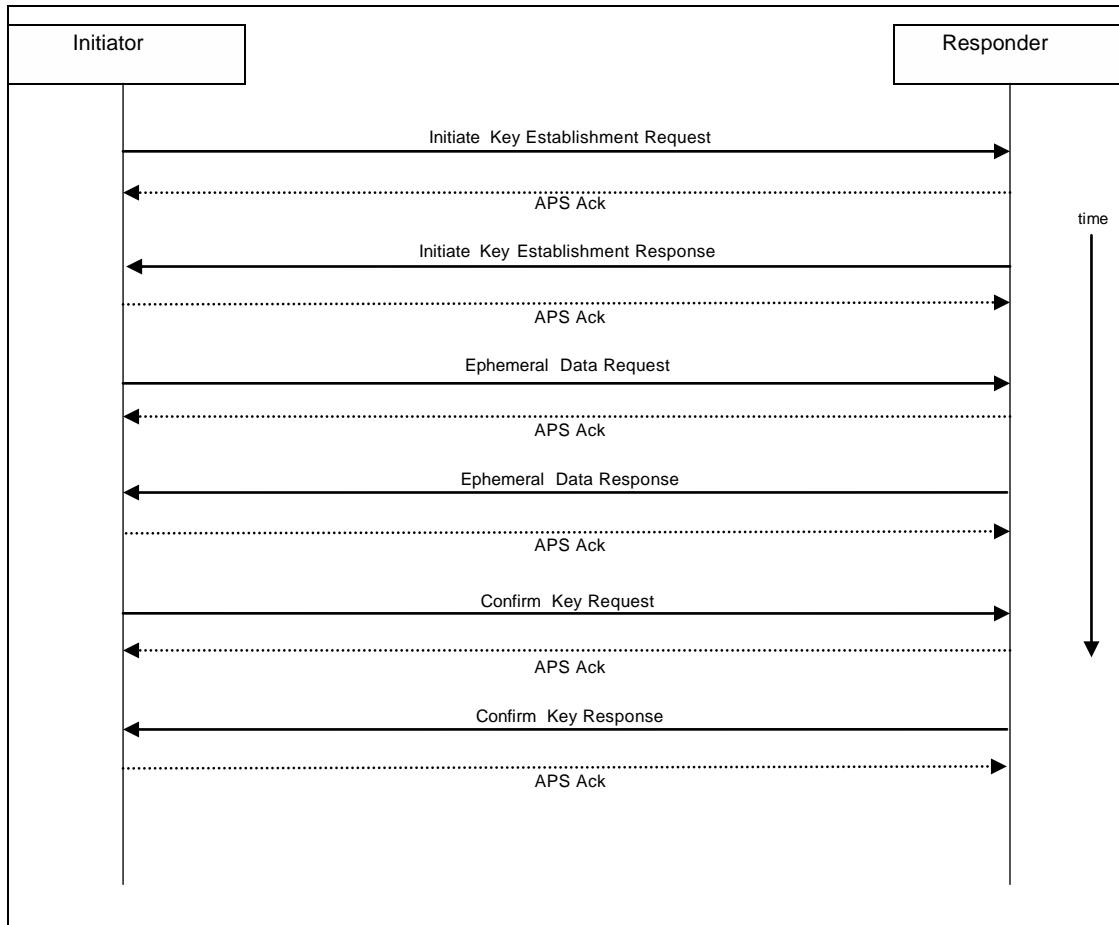


Figure C-12– Key Establishment Command Exchange

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	00

4275 APS Header

Frame Control	0x40
Destination Endpoint	0x0A
Cluster Identifier	0x0800
Profile ID	0x0109
Source Endpoint	0x0A
APS Counter	0x01

4276

4277

ZCL Header

Frame Control	0x01	Client to Server
Sequence Number	0x00	
Command Identifier	0x00	<i>Initiate Key Establishment Request</i>
Key Establishment Suite	0x0001	ECMQV
Ephemeral Data Generate Time	0x03	
Confirm Key Generate Time	0x06	
Identity (IDU)	*	Device 2's certificate

4278

C.5.2.2 Initiate Key Establishment Response

4280 The following is the APS message sent by the responder (device 1) to the initiator (device 2) for
 4281 the initiate key establishment response.

4282 40 0A 00 08 09 01 0A 01
 4283 09 00 00 01 00 03 06 03
 4284 04 5F DF C8 D8 5F FB 8B
 4285 39 93 CB 72 DD CA A5 5F
 4286 00 B3 E8 7D 6D 00 00 00
 4287 00 00 00 00 01 54 45 53
 4288 54 53 45 43 41 01 09 00
 4289 06 00 00 00 00 00 00

4290

APS Header

Frame Control	0x40
Destination Endpoint	0x0A
Cluster Identifier	0x0800
Profile ID	0x0109
Source Endpoint	0x0A
APS Counter	0x01

4291

4292

ZCL Header

Frame Control	0x09	Server to Client
Sequence Number	0x00	
Command Identifier	0x00	<i>Initiate Key Establishment Response</i>
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	<i>Ephemeral Data Request</i>
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

4306 **C.5.2.4 Ephemeral Data Response**

4307 The following is the APS message sent by the responder to the initiator for the ephemeral data
4308 response.

4309 40 0A 00 08 09 01 0A 02
4310 09 01 01 03 06 AB 52 06
4311 22 01 D9 95 B8 B8 59 1F
4312 3F 08 6A 3A 2E 21 4D 84
4313 5E

4314 **APS Header**

Frame Control	0x40
Destination Endpoint	0x0A
Cluster Identifier	0x0800
Profile ID	0x0109
Source Endpoint	0x0A
APS Counter	0x02

4315

4316 **ZCL Header**

Frame Control	0x09	Server to Client
Sequence Number	0x01	
Command Identifier	0x01	<i>Ephemeral Data Response</i>
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	

4317

4318 **C.5.2.5 Confirm Key Request**

4319 The following is the APS message sent by the initiator to the responder for the confirm key
4320 request.

4321 40 0A 00 08 09 01 0A 03
4322 01 02 02 B8 2F 1F 97 74
4323 74 0C 32 F8 0F CF C3 92
4324 1B 64 20

4325 **APS Header**

4326

Frame Control	0x40
Destination Endpoint	0x0A
Cluster Identifier	0x0800
Profile ID	0x0109
Source Endpoint	0x0A
APS Counter	0x02

4327

ZCL Header

Frame Control	0x01	Client to Server
Sequence Number	0x02	
Command Identifier	0x02	<i>Confirm Key Request</i>
Secure Message Authentication Code (MACU)	B8 2F 1F 97 74 74 0C 32 F8 0F CF C3 92 1B 64 20	

4328

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 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

ZCL Header

Frame Control	0x09	Server to Client
Sequence Number	0x02	

Command Identifier	0x02	<i>Confirm Key Response</i>
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.

4351 **C.5.3.2 Key Derivation Function (KDF)**

4352 Once a shared secret (Z) is established, a transform is done to create a SMAC (Secure
4353 Message Authentication Code) and a shared ZigBee Key.

4354 **C.5.3.3 Initiator Transform**

4355 Upon receipt of the responder's ephemeral data response, the initiator has all the data necessary
4356 to calculate the shared secret and derive the data for the confirm key request (SMAC).

4357 **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

4358

4359 **C.5.3.3.2 Step Summary**

4360 **1** Derive the Shared Secret using the ECMQV primitives
4361 a Z = ECC_GenerateSharedSecret(Private(U), E(U), E-P(U), Cert(V), E(V), CA)

4362 **2** Derive the Keying data

4363 a Hash-1 = Z || 00 00 00 01 || SharedData
4364 b Hash-2 = Z || 00 00 00 02 || SharedData

4365 **3** Parse KeyingData as follows

4366 a MacKey = First 128 bits (Hash-1) of KeyingData
4367 b KeyData = Second 128 bits (Hash-2) of KeyingData

4368 **4** Create MAC(U)

4369 a MAC(U) = MAC(MacKey) { M(U) || ID(U) || ID(V) || E(U) || E(V) }

4370 **5** Send MAC(U) to V.

4371 **6** Receive MAC(V) from V.

4372 **7** Calculate MAC(V)'

4373 a MAC(V) = MAC(MacKey) { M(V) || ID(V) || ID(U) || E(V) || E(U) }

4374 **8** Verify MAC(V)' is the same as MAC(V).

4375 **C.5.3.3.3 Detailed Steps**

4376 **1** Derive the Shared Secret using the ECMQV primitives

4377 a Z = ECC_GenerateSharedSecret(Private(U), E(U), E-P(U), Cert(V), E(V), CA)
4378 00 E0 D2 C3 CC D5 C1 06 A8 9C 4F 6C C2 6A 5F 7E

- 4379 C9 DF 78 A7 BE
- 4380 **2 Derive the Keying data**
- 4381 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
- 4387 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
- 4394 **3 Parse KeyingData as follows**
- 4395 a MacKey = First 128 bits (Hash-1) of KeyingData
- 4396 b KeyData = Second 128 bits (Hash-2) of KeyingData
- 4397 **4 Create MAC(U)**
- 4398 a MAC(U) = MAC(MacKey) { M(U) || ID(U) || ID(V) || E(U) || E(V) }
Concatenation
02 00 00 00 00 00 00 00 00 02 00 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
- 4406 **5 Send MAC(U) to V.**
- 4407 **6 Receive MAC(V) from V.**
- 4408 **7 Calculate MAC(V)'**
- 4409 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 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
- 4417 **8 Verify MAC(V)' is the same as MAC(V).**

4418 **C.5.3.4 Responder Transform**

4419 Upon receipt of the initiator's confirm key request, the responder has all the data necessary to
4420 calculate the shared secret, validate the initiator's confirm key message, and derive the data
4421 for the confirm key response (SMAC).

4422 **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

4423

4424 **C.5.3.4.2 Step Summary**

- 4425 **1** Derive the Shared Secret using the ECMQV primitives
4426 a $Z = \text{ECC_GenerateSharedSecret}(\text{Private}(V), E(V), E-P(V), \text{Cert}(U), E(U), CA)$
- 4427 **2** Derive the Keying data
4428 a $\text{Hash-1} = Z \parallel 00\ 00\ 00\ 01 \parallel \text{SharedData}$
4429 b $\text{Hash-2} = Z \parallel 00\ 00\ 00\ 02 \parallel \text{SharedData}$

4430 **3** Parse KeyingData as follows

- 4431 a $\text{MacKey} = \text{First 128 bits (Hash-1) of KeyingData}$
4432 b $\text{KeyData} = \text{Second 128 bits (Hash-2) of KeyingData}$

4433 **4** Create MAC(V)

- 4434 a $\text{MAC}(V) = \text{MAC}(\text{MacKey}) \{ M(V) \parallel ID(V) \parallel ID(U) \parallel E(V) \parallel E(U) \}$

4435 **5** Calculate MAC(U)'

- 4436 a $\text{MAC}(U) = \text{MAC}(\text{MacKey}) \{ M(U) \parallel ID(U) \parallel ID(V) \parallel E(U) \parallel E(V) \}$

4437 **6** Verify $\text{MAC}(U)'$ is the same as $\text{MAC}(U)$.

4438 **7** Send $\text{MAC}(V)$ to U.

4439 **C.5.3.4.3 Detailed Steps**

- 4440 **1** Derive the Shared Secret using the ECMQV primitives
4441 a $Z = \text{ECC_GenerateSharedSecret}(\text{Private}(U), E(U), E-P(U), \text{Cert}(V), E(V), CA)$
4442 00 E0 D2 C3 CC D5 C1 06 A8 9C 4F 6C C2 6A 5F 7E
4443 C9 DF 78 A7 BE
- 4444 **2** Derive the Keying data
4445 a $\text{Hash-1} = Z \parallel 00\ 00\ 00\ 01 \parallel \text{SharedData}$
4446 **Concatenation**

4447 00 E0 D2 C3 CC D5 C1 06 A8 9C 4F 6C C2 6A 5F 7E
 4448 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

4457 3 Parse KeyingData as follows

4458 a MacKey = First 128 bits (Hash-1) of KeyingData

4459 b KeyData = Second 128 bits (Hash-2) of KeyingData

4460 4 Create MAC(V)

4461 a MAC(V) = MAC(MacKey) { M(V) || ID(V) || ID(U) || E(V) || E(U) }
Concatenation

03 00 00 00 00 00 00 01 00 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

4469 5 Calculate MAC(V)'

4470 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

4478 6 Verify MAC(V)' is the same as MAC(V).

4479 7 Send MAC(V) to U.

4480

4481 **C.6 Key Establishment Test Vectors for Cryptographic Suite 2**

4482 The following details the key establishment exchange data transformation and validation of
4483 test vectors for a pair of Smart Energy devices using Certificate based key exchange (CBKE)
4484 using Elliptical Curve Cryptography (ECC).

4485 **C.6.1 Preconfigured Data**

4486 Each device is expected to have been preinstalled with security information prior to initiating
4487 key establishment. The preinstalled data consists of the Certificate Authority's Public Key, a
4488 device specific certificate, and a device specific private key.

4489 **C.6.1.1 CA Public Key**

4490 The following is the Certificate Authority's Public Key:

4491 02 07 A4 45 02 2D 9F 39 f4 9B DC 38 38 00 26 A2
4492 7A 9E 0A 17 99 31 3A B2 8C 5C 1A 1C 6B 60 51 54
4493 DB 1D FF 67 52

4494 **C.6.1.2 Responder Data**

4495 The following is the certificate for device 1. The device has an EUI-64 address of
4496 0A:0B:0C:0D:0E:0F:10:11, and will be the responder.

4497

4498 Certificate:

4499 00 26 22 A5 05 E8 93 8F 27 0D 08 11 12 13 14 15
4500 16 17 18 00 52 92 A3 5B FF FF FF FF 0A 0B 0C 0D
4501 0E 0F 10 11 88 03 03 B4 E9 DC 54 3A 64 33 3C 98
4502 23 08 02 2B 54 E6 7E 2F 15 F5 32 55 1B 0A 11 E2
4503 E2 C1 C1 D3 09 7A 43 24 E7 ED

4504

4505 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

4506

4507 The private key for device 1 is as follows:

4508 01 51 CD 0D BC B8 04 74 BF 7A C9 FE EB E3 9C 7A
 4509 32 A6 35 18 93 8F CA 97 54 AA E1 32 BC 9C 73 BE
 4510 94 A7 E1 BE

4511 The public key for device 1 is as follows:

4512 02 02 F4 FA 2A 30 40 43 3C 68 20 29 9D 18 2A 10
 4513 42 E4 14 04 E3 37 C5 7F 47 71 6B 42 DF AF 97 0F
 4514 15 80 A0 4C 9B
 4515

4516 C.6.1.3 Initiator Data

4517 The following is the certificate for device 2. The device has an EUI-64 address of
 4518 0A:0B:0C:0D:0E:0F:10:12, and will be the initiator.

4519

4520 Certificate:

4521 00 84 A9 33 B3 7F 01 8D EC 0D 08 11 12 13 14 15
 4522 16 17 18 00 52 92 A3 8A FF FF FF FF 0A 0B 0C 0D
 4523 0E 0F 10 12 88 03 07 62 77 E2 F7 E2 25 2B 16 A0
 4524 E9 2B 6E 87 71 BB 3F 20 79 46 CB D4 A4 5D 9A 9D
 4525 F6 ED AB 8C 79 6A 48 E8 9D EC
 4526

4527 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

4528

4529 The private key for device 2 is as follows:

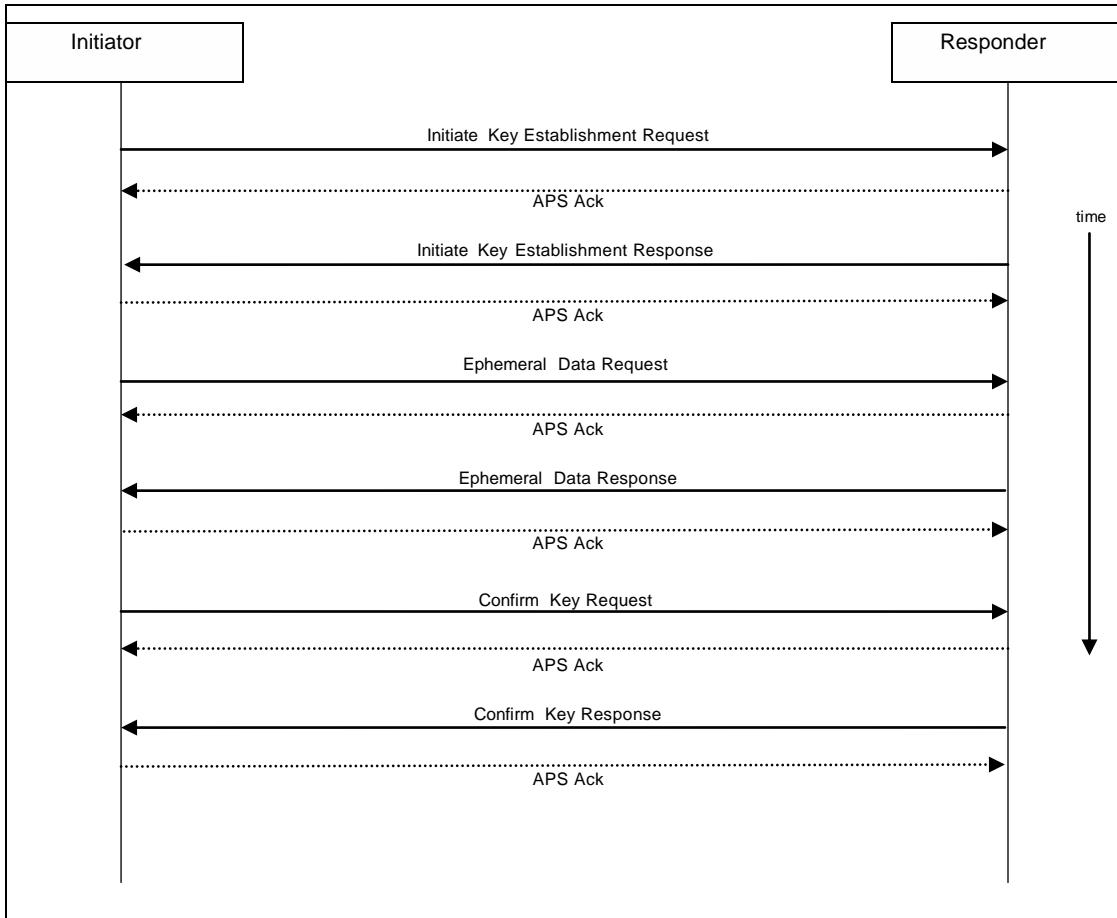
4530 00 F2 56 1A DB 39 EF 49 C1 D6 2E F5 18 6C 6E 0C
 4531 15 8A 5A 45 BF CE 38 66 09 31 AC C3 69 45 92 D5
 4532 AC DE 90 06

4533 The public key for device 2 is as follows:

4534 03 03 0E 56 F7 AD E8 66 E7 63 72 76 4B A2 0A 9F
4535 F1 FE 4C AE 52 2F 94 83 9E 70 F2 AD FC 1C A3 E9
4536 7F 4D DC AF 2E
4537

4538 C.6.2 Key Establishment Messages

4539 The following is the basic flow of messages back and forth between the initiator and the
4540 responder performing key establishment using the Key Establishment Cluster.



4541 **Figure C-13– Key Establishment Command Exchange**

4542 C.6.2.1 Initiate Key Establishment Request

4543 The following is the APS message sent by the initiator (device 2) to the responder (device 1) for
4544 the initiate key establishment request.

4545 40 0A 00 08 09 01 0A 01 01 00 00 02 00 03 06 00
4546 84 A9 33 B3 7F 01 8D EC 0D 08 11 12 13 14 15 16
4547 17 18 00 52 92 A3 8A FF FF FF FF 0A 0B 0C 0D 0E

4548 0F 10 12 88 03 07 62 77 E2 F7 E2 25 2B 16 A0 E9
 4549 2B 6E 87 71 BB 3F 20 79 46 CB D4 A4 5D 9A 9D F6
 4550 ED AB 8C 79 6A 48 E8 9D EC

4551 APS Header

Frame Control	0x40
Destination Endpoint	0x0A
Cluster Identifier	0x0800
Profile ID	0x0109
Source Endpoint	0x0A
APS Counter	0x01

4552
4553

ZCL Header

Frame Control	0x01	Client to Server
Sequence Number	0x00	
Command Identifier	0x00	<i>Initiate Key Establishment Request</i>
Requested Security Suite	0x0002	CBKE-ECMQV-V2
Ephemeral Data Generate Time	0x03	
Confirm Key Generate Time	0x06	
Identity (IDU)	*	Device 2's certificate

4554

4555 C.6.2.2 Initiate Key Establishment Response

4556 The following is the APS message sent by the responder (device 1) to the initiator (device 2) for
 4557 the initiate key establishment response.

4558 40 0A 00 08 09 01 0A 01 09 00 00 02 00 03 06 00
 4559 26 22 A5 05 E8 93 8F 27 0D 08 11 12 13 14 15 16
 4560 17 18 00 52 92 A3 5B FF FF FF FF 0A 0B 0C 0D 0E
 4561 0F 10 11 88 03 03 B4 E9 DC 54 3A 64 33 3C 98 23
 4562 08 02 2B 54 E6 7E 2F 15 F5 32 55 1B 0A 11 E2 E2
 4563 C1 C1 D3 09 7A 43 24 E7 ED

4564 APS Header

Frame Control	0x40
Destination Endpoint	0x0A
Cluster Identifier	0x0800
Profile ID	0x0109
Source Endpoint	0x0A
APS Counter	0x01

4565

4566

ZCL Header

Frame Control	0x09	Server to Client
Sequence Number	0x00	
Command Identifier	0x00	<i>Initiate Key Establishment Response</i>
Accepted Security Suite	0x0002	CBKE-ECMQV-V2
Ephemeral Data Generate Time	0x03	
Confirm Key Generate Time	0x06	
Identity (IDV)	*	Device 1's certificate

4567

C.6.2.3 Ephemeral Data Request

4569 The following is the APS message sent by the initiator to the responder for the ephemeral data request.

4570
4571 40 0A 00 08 09 01 0A 02 01 01 01 03 05 F3 39 4E
4572 15 68 06 60 EE CA A3 67 88 D9 B6 F3 12 B9 71 CE
4573 2C 96 17 57 0B F7 DF CD 21 C9 72 01 77 62 C3 32

4574

APS Header

Frame Control	0x40
Destination Endpoint	0x0A
Cluster Identifier	0x0800
Profile ID	0x0109
Source Endpoint	0x0A
APS Counter	0x02

4575

ZCL Header

Frame Control	0x01	Client to Server
Sequence Number	0x01	
Command Identifier	0x01	<i>Ephemeral Data Request</i>
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

4577

C.6.2.4 Ephemeral Data Response

4579 The following is the APS message sent by the responder to the initiator for the ephemeral data response.

4581 40 0A 00 08 09 01 0A 02 09 01 01 03 00 9A 51 31

4582 CF 5B 92 A0 16 37 8C 0F 7F 28 4E CD 47 F9 40 10
 4583 F8 75 D4 3B F1 E9 A6 54 74 AD BF C6 36 96 A9 30

4584 **APS Header**

Frame Control	0x40
Destination Endpoint	0x0A
Cluster Identifier	0x0800
Profile ID	0x0109
Source Endpoint	0x0A
APS Counter	0x02

4585

4586 **ZCL Header**

Frame Control	0x09	Server to Client
Sequence Number	0x01	
Command Identifier	0x01	<i>Ephemeral Data Response</i>
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

4587

4588 **C.6.2.5 Confirm Key Request**

4589 The following is the APS message sent by the initiator to the responder for the confirm key
 4590 request.

4591 40 0A 00 08 09 01 0A 03
 4592 01 02 02 BF 7E 1A 26 D4
 4593 EF 70 38 B5 68 13 E4 65
 4594 A1 31 C9

4595 **APS Header**

Frame Control	0x40
Destination Endpoint	0x0A
Cluster Identifier	0x0800
Profile ID	0x0109
Source Endpoint	0x0A
APS Counter	0x03

4596

4597 **ZCL Header**

Frame Control	0x01	Client to Server
Sequence Number	0x02	
Command Identifier	0x02	<i>Confirm Key Request</i>
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	<i>Confirm Key Response</i>
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)

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

4612

4613 *Note: '||' stands for bitwise concatenation*4614 **C.6.3.1 ECMQV Primitives**

4615 It is assumed that an ECC library is available for creating the shared secret given the local
 4616 private key, local ephemeral public & private key, remote device's certificate, remote
 4617 device's ephemeral public key, and the certificate authority's public key. Further it is assumed
 4618 that this library has been separately validated with a set of ECC test vectors. Those test
 4619 vectors are outside the scope of this document.

4620 **C.6.3.2 Key Derivation Function (KDF)**

4621 Once a shared secret (Z) is established, a transform is done to create a SMAC (Secure
 4622 Message Authentication Code) and a shared ZigBee Key.

4623 **C.6.3.3 Initiator Transform**

4624 Upon receipt of the responder's ephemeral data response, the initiator has all the data necessary
 4625 to calculate the shared secret and derive the data for the confirm key request (SMAC).

4626 **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

4627

4628 C.6.3.3.2 **Step Summary**

- 4629 **1** Derive the Shared Secret using the ECMQV primitives
 - 4630 a $Z = \text{ECC_GenerateSharedSecret}(\text{Private}(U), E(U), E-P(U), \text{Cert}(V), E(V), CA)$
- 4631 **2** Derive the Keying data
 - 4632 a $\text{Hash-1} = Z \parallel 00\ 00\ 00\ 01 \parallel \text{SharedData}$
 - 4633 b $\text{Hash-2} = Z \parallel 00\ 00\ 00\ 02 \parallel \text{SharedData}$
- 4634 **3** Parse KeyingData as follows
 - 4635 a $\text{MacKey} = \text{First 128 bits (Hash-1) of KeyingData}$
 - 4636 b $\text{KeyData} = \text{Second 128 bits (Hash-2) of KeyingData}$
- 4637 **4** Create MAC(U)
 - 4638 a $\text{MAC}(U) = \text{MAC}(\text{MacKey}) \{ M(U) \parallel ID(U) \parallel ID(V) \parallel E(U) \parallel E(V) \}$
- 4639 **5** Send MAC(U) to V.
- 4640 **6** Receive MAC(V) from V.
- 4641 **7** Calculate MAC(V)'
 - 4642 a $\text{MAC}(V) = \text{MAC}(\text{MacKey}) \{ M(V) \parallel ID(V) \parallel ID(U) \parallel E(V) \parallel E(U) \}$
- 4643 **8** Verify MAC(V)' is the same as MAC(V).

4644
4645 C.6.3.3.3 **Detailed Steps**

- 4646 **1** Derive the Shared Secret using the ECMQV primitives
 - 4647 a $Z = \text{ECC_GenerateSharedSecret}(\text{Private}(U), E(U), E-P(U), \text{Cert}(V), E(V), CA)$

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	00	00	00	01								
 - 4651 **2** Derive the Keying data
 - 4652 a $\text{Hash-1} = Z \parallel 00\ 00\ 00\ 01 \parallel \text{SharedData}$

Concatenation

 - 4654 04 F7 72 4A 9A 77 B2 1D 27 47 CC EF 68 A4 57 E4
 - 4655 52 46 C4 BE 9F 66 FD 94 25 22 7B CB 2C C5 18 0E
 - 4656 A9 CC CB 9A 00 00 00 01

Hash

 - 4658 ED 38 0A 00 29 66 00 FB 6B 89 30 25 DE 5F D1 37
 - 4660 **b** $\text{Hash-2} = Z \parallel 00\ 00\ 00\ 02 \parallel \text{SharedData}$
- Concatenation**

4662 04 F7 72 4A 9A 77 B2 1D 27 47 CC EF 68 A4 57 E4
 4663 52 46 C4 BE 9F 66 FD 94 25 22 7B CB 2C C5 18 0E
 4664 A9 CC CB 9A 00 00 00 02

4665 **Hash**
 4666 AA 46 89 C7 0B E0 FA F0 C9 BE 53 4A BD 9F 4C DC
 4667

- 4668 **3** Parse KeyingData as follows
 4669 a MacKey = First 128 bits (Hash-1) of KeyingData
 4670 b KeyData = Second 128 bits (Hash-2) of KeyingData

- 4671 **4** Create MAC(U)
 4672 a MAC(U) = MAC(MacKey) { M(U) || ID(U) || ID(V) || E(U) || E(V) }

4673 **Concatenation**
 4674 02 0a 0b 0c 0d 0e 0f 10 12 0a 0b 0c 0d 0e 0f 10
 4675 11 03 05 F3 39 4E 15 68 06 60 EE CA A3 67 88 D9
 4676 B6 F3 12 B9 71 CE 2C 96 17 57 0B F7 DF CD 21 C9
 4677 72 01 77 62 C3 32 03 00 9A 51 31 CF 5B 92 A0 16
 4678 37 8C 0F 7F 28 4E CD 47 F9 40 10 F8 75 D4 3B F1
 4679 E9 A6 54 74 AD BF C6 36 96 A9 30

4680 **Hash**
 4681 BF 7E 1A 26 D4 EF 70 38 B5 68 13 E4 65 A1 31 C9
 4682

- 4683 **5** Send MAC(U) to V.

- 4684 **6** Receive MAC(V) from V.

- 4685 **7** Calculate MAC(V)'
 4686 a MAC(V) = MAC(MacKey) { M(V) || ID(V) || ID(U) || E(V) || E(U) }
 4687 **Concatenation**
 4688 03 0a 0b 0c 0d 0e 0f 10 11 0a 0b 0c 0d 0e 0f 10
 4689 12 03 00 9A 51 31 CF 5B 92 A0 16 37 8C 0F 7F 28
 4690 4E CD 47 F9 40 10 F8 75 D4 3B F1 E9 A6 54 74 AD
 4691 BF C6 36 96 A9 30 03 05 F3 39 4E 15 68 06 60 EE
 4692 CA A3 67 88 D9 B6 F3 12 B9 71 CE 2C 96 17 57 0B
 4693 F7 DF CD 21 C9 72 01 77 62 C3 32

4694 **Hash**
 4695 C5 B4 32 A9 99 5A 09 2F 44 49 F8 36 13 93 00 64
 4696

- 4697 **8** Verify MAC(V)' is the same as MAC(V).
 4698

4699 C.6.3.4 Responder Transform

4700 Upon receipt of the initiator's confirm key request, the responder has all the data necessary to
4701 calculate the shared secret, validate the initiator's confirm key message, and derive the data
4702 for the confirm key response (SMAC).

4703 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

4704

4705 C.6.3.4.2 Step Summary

4706 1 Derive the Shared Secret using the ECMQV primitives

4707 a $Z = \text{ECC_GenerateSharedSecret}(\text{Private}(V), E(V), E-P(V), \text{Cert}(U), E(U), CA)$

4708 2 Derive the Keying data

4709 a $\text{Hash-1} = Z \parallel 00\ 00\ 00\ 01 \parallel \text{SharedData}$

4710 b $\text{Hash-2} = Z \parallel 00\ 00\ 00\ 02 \parallel \text{SharedData}$

4711 3 Parse KeyingData as follows

4712 a $\text{MacKey} = \text{First 128 bits (Hash-1) of KeyingData}$

4713 b $\text{KeyData} = \text{Second 128 bits (Hash-2) of KeyingData}$

4714 4 Create MAC(V)

4715 a $\text{MAC}(V) = \text{MAC}(\text{MacKey}) \{ M(V) \parallel ID(V) \parallel ID(U) \parallel E(V) \parallel E(U) \}$

4716 5 Calculate MAC(U)'

4717 a $\text{MAC}(U) = \text{MAC}(\text{MacKey}) \{ M(U) \parallel ID(U) \parallel ID(V) \parallel E(U) \parallel E(V) \}$

4718 6 Verify $\text{MAC}(U)'$ is the same as $\text{MAC}(U)$.

4719 7 Send $\text{MAC}(V)$ to U

4720

4721 C.6.3.4.3 Detailed Steps

4722 1 Derive the Shared Secret using the ECMQV primitives

4723 a $Z = \text{ECC_GenerateSharedSecret}(\text{Private}(V), E(V), E-P(V), \text{Cert}(U), E(U), CA)$

4724 04 F7 72 4A 9A 77 B2 1D 27 47 CC EF 68 A4 57 E4

4725 52 46 C4 BE 9F 66 FD 94 25 22 7B CB 2C C5 18 0E

4726 A9 CC CB 9A

4727 **2 Derive the Keying data**

4728 a Hash-1 = Z || 00 00 00 01 || SharedData

4729 **Concatenation**

4730 04 F7 72 4A 9A 77 B2 1D	27 47 CC EF 68 A4 57 E4
4731 52 46 C4 BE 9F 66 FD 94	25 22 7B CB 2C C5 18 0E
4732 A9 CC CB 9A 00 00 00 01	

4733 **Hash**

4734 ED 38 0A 00 29 66 00 FB	6B 89 30 25 DE 5F D1 37
----------------------------------	-------------------------

4736 b Hash-2 = Z || 00 00 00 02 || SharedData

4737 **Concatenation**

4738 04 F7 72 4A 9A 77 B2 1D	27 47 CC EF 68 A4 57 E4
4739 52 46 C4 BE 9F 66 FD 94	25 22 7B CB 2C C5 18 0E
4740 A9 CC CB 9A 00 00 00 02	

4741 **Hash**

4742 AA 46 89 C7 0B E0 FA F0	C9 BE 53 4A BD 9F 4C DC
----------------------------------	-------------------------

4744 **3 Parse KeyingData as follows**

4745 a MacKey = First 128 bits (Hash-1) of KeyingData

4746 b KeyData = Second 128 bits (Hash-2) of KeyingData

4747 **4 Create MAC(V)**

4748 a MAC(V) = MAC(MacKey) { M(V) || ID(V) || ID(U) || E(V) || E(U) }

4749 **Concatenation**

4750 03 0a 0b 0c 0d 0e 0f 10	11 0a 0b 0c 0d 0e 0f 10
4751 12 03 00 9A 51 31 CF 5B	92 A0 16 37 8C 0F 7F 28
4752 4E CD 47 F9 40 10 F8 75	D4 3B F1 E9 A6 54 74 AD
4753 BF C6 36 96 A9 30 03 05	F3 39 4E 15 68 06 60 EE
4754 CA A3 67 88 D9 B6 F3 12	B9 71 CE 2C 96 17 57 0B
4755 F7 DF CD 21 C9 72 01 77	62 C3 32

4756 **Hash**

4757 C5 B4 32 A9 99 5A 09 2F	44 49 F8 36 13 93 00 64
----------------------------------	-------------------------

4759 **5 Calculate MAC(U)'**

4760 a MAC(U)' = MAC(MacKey) { M(U) || ID(U) || ID(V) || E(U) || E(V) }

4761 **Concatenation**

4762 02 0a 0b 0c 0d 0e 0f 10	12 0a 0b 0c 0d 0e 0f 10
4763 11 03 05 F3 39 4E 15 68	06 60 EE CA A3 67 88 D9
4764 B6 F3 12 B9 71 CE 2C 96	17 57 0B F7 DF CD 21 C9
4765 72 01 77 62 C3 32 03 00	9A 51 31 CF 5B 92 A0 16
4766 37 8C 0F 7F 28 4E CD 47	F9 40 10 F8 75 D4 3B F1
4767 E9 A6 54 74 AD BF C6 36	96 A9 30

4768 **Hash**
4769 BF 7E 1A 26 D4 EF 70 38 B5 68 13 E4 65 A1 31 C9
4770
4771 **6** Verify MAC(U)' is the same as MAC(U).
4772 **7** Send MAC(V) to U
4773

4774

Annex D SMART ENERGY CLUSTER DESCRIPTIONS

4775

4776 The candidate material in this annex describing the Smart Energy Clusters, when approved, will
4777 be merged into the Foundation document of the ZigBee Cluster Library (ZCL) by the Cluster
4778 Library Development Board.

4779

D.1 Annex Guidelines

4780

D.1.1 Client/Server Model Information

4781 The ZigBee Cluster Library Specification is used as the guiding reference for defining the
4782 rule set in defining the Client/Server model for the Smart Energy Profile. Please note the
4783 following items influence the further refinement of that definition:

4784 • Attributes can be defined for both Client and Server side clusters. Attributes can be used to
4785 understand current state of activities within a device, enhancing both the diagnostic and
4786 maintenance of devices or the processes supported by that device.

4787 • The ESI device acts as the transition point from upstream Wide Area Network (and
4788 subsequent upstream systems) to the ZigBee network. Because of this responsibility, in some
4789 of the clusters it acts as a proxy for the upstream systems. In situations in which the proxy
4790 condition occurs, plus where attributes are defined or commands (transactions) are initiated
4791 on both client/ server sides, the ESI will be by default labeled as the Server side in the cluster
4792 descriptions.

4793

D.1.2 Interpretation of Reserved Field Values or Bitmaps

4794 To support backwards compatibility, devices should ignore any values or bit settings for any
4795 reserved field values. If the field is necessary for interpretation or in conjunction with other fields
4796 the whole message can be ignored.

4797 To enable future growth and ensure backwards compatibility, any existing devices which
4798 encounter any fields applied after the end of a command shall treat them as reserved fields. The
4799 future addition of fields applied after the end of defined cluster commands are reserved
4800 solely for ZigBee specifications, Manufacturers shall not add fields after the end of commands.

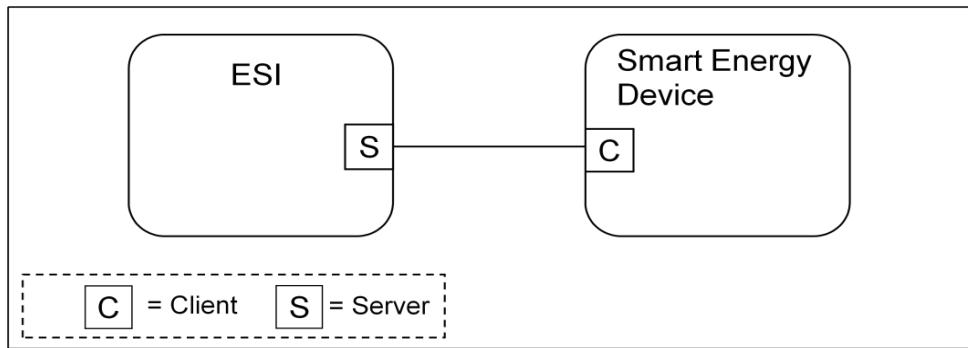
4801

4802 **D.2 Demand Response and Load Control Cluster**

4803 **D.2.1 Overview**

4804 This cluster provides an interface to the functionality of Smart Energy Demand Response and
4805 Load Control. Devices targeted by this cluster include thermostats and devices that support load
4806 control.

4807



4808
4809 **Figure D-1– Demand Response/Load Control Cluster Client Server Example**

4810 Please note the ESI is defined as the Server due to its role in acting as the proxy for upstream
4811 demand response/load control management systems and subsequent data stores.

4812 **D.2.2 Server**

4813 By default the ESI will be labeled as the Server side in the cluster descriptions, being able to
4814 initiate load control commands to other devices in the network.

4815 **D.2.2.1 Dependencies**

4816 A server device shall be capable of storing at least two load control events.

4817 Events carried using this cluster include a timestamp with the assumption that target devices
4818 maintain a real-time clock. Devices can acquire and synchronize their internal clocks with the
4819 ESI as described in sub-clause 5.12.1.1.

4820 If a device does not support a real-time clock, it is assumed the device will ignore all values within
4821 the Time field except the "Start Now" value.

4822 Additionally, for devices without a real-time clock, it is assumed those devices will utilize a
4823 method (i.e. ticks, countdowns, etc.) to approximate the correct duration period.

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	<i>Load Control Event</i>	M
0x01	<i>Cancel Load Control Event</i>	M
0x02	<i>Cancel All Load Control Events</i>	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)	Criticality 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 Temperature Set Point (O)	Average Load Adjustment Percentage (O)	Duty Cycle (O)	Event Control (M)

4835

Figure D-2– Load Control Event Command Payload

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.

4858 **Utility Enrollment Group (mandatory):** The Utility Enrollment Group field can be used in
 4859 conjunction with the Device Class bits. It provides a mechanism to direct Load Control Events
 4860 to groups of Devices. Example, by assigning two different groups relating to either Demand
 4861 Response programs or geographic areas, Load Control Events can be further directed for a sub-
 4862 set of Device Classes (i.e. Device Class Bit 0 and Utility Enrollment Group #1 vs. Device
 4863 Class Bit0 and Utility Enrollment Group #2). 0x00 addresses all groups, and values 0x01 to
 4864 0xFF address individual groups that match. Please refer to sub-clause D.2.3.2.1 for further
 4865 details.

4866 If the Device Class and/or Utility Enrollment Group fields don't apply to your End Device, the
 4867 *Load Control Event* command shall be ignored by either dropping the message and not replying
 4868 at all or by sending back a Default Response message with a SUCCESS status code.

4869 **Start Time (mandatory):** UTC Timestamp representing when the event is scheduled to start. A
 4870 start time of 0x00000000 is a special time denoting "now." If the device would send an event
 4871 with a Start Time of now, adjust the Duration In Minutes field to correspond to the remainder of
 4872 the event.

4873 **Duration In Minutes (mandatory):** Duration of this event in number of minutes. Maximum
 4874 value is 1440 (one day).

4875 **Criticality Level (mandatory):** This field defines the level of criticality of this event. The
 4876 action taken by load control devices for an event can be solely based on this value, or
 4877 combination with other Load Control Event fields supported by this device. For example,
 4878 additional fields such as Average Load Adjustment Percentage, Duty Cycle, Cooling
 4879 Temperature Offset, Heating Temperature Offset, Cooling Temperature Set Point or Heating
 4880 Temperature Set Point can be used in combination with the Criticality level. Criticality levels
 4881 are listed in Table D-3.

4882

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

9	Service Disconnect	Mandatory
0x0A to 0x0F	Utility Defined	Utility Defined
0x10 to 0xFF	Reserved	

4883

4884 The criticality level 0x0 and 0x10 to 0xFF are reserved for future profile changes and not used.

4885 “Green” event, level 0x01, may be used to denote that the energy delivered uses an abnormal
4886 amount from non-“green” sources. Participation in this event is voluntary.

4887 The criticality levels 0x02 through 0x06 (Levels 1 through 5) indicate progressively
4888 increasing levels of load reduction are being requested by the utility. Participation in these events
4889 is voluntary.

4890 The criticality level 0x07 is used to indicate an “Emergency” event. Participation in this event is
4891 mandatory, as defined by the utility. The expected response to this event is termination of all
4892 non-essential energy use, as defined by the utility. Exceptions to participation in this event type
4893 must be managed by the utility.

4894 The criticality level 0x08 is used to indicate a “Planned Outage” event. Participation in
4895 this event is mandatory, as defined by the utility. The expected response to this event is
4896 termination of delivery of all non-essential energy, as defined by the utility. Exceptions to
4897 participation in this event type must be managed by the utility.

4898 The criticality level 0x09 is used to indicate a “Service Disconnect” event. Participation in this
4899 event is mandatory, as defined by the utility. The expected response to this event is
4900 termination of delivery of all non-essential energy, as defined by the utility. Exceptions to
4901 participation in this event type must be managed by the utility.

4902 Levels 0x0A to 0x0F are available for Utility Defined criticality levels.

4903 **Cooling Temperature Offset (optional):** Requested offset to apply to the normal cooling
4904 setpoint at the time of the start of the event in + 0.1 °C.

4905 **Heating Temperature Offset (optional):** Requested offset to apply to the normal heating
4906 setpoint at the time of the start of the event in + 0.1 °C.

4907 The Cooling and Heating Temperature Offsets represent a temperature change (Delta
4908 Temperature) that will be applied to both the associated heating and cooling set points. The
4909 temperature offsets (Delta Temperatures) will be calculated per the Local Temperature in the
4910 Thermostat. The calculated temperature will be interpreted as the number of degrees to be
4911 added to the cooling set point and subtracted from the heating set point. Sequential demand
4912 response events are not cumulative. The Offset shall be applied to the normal setpoint.

- 4913 Each offset represents the temperature offset (Delta Temperature) in degrees Celsius, as
4914 follows: Delta Temperature Offset / 10 = delta temperature in degrees Celsius. Where $0.00^{\circ}\text{C} \leq$
4915 temperature $\leq 25.4^{\circ}\text{C}$, corresponding to a Temperature in the range 0x00 to 0x0FE. The
4916 maximum resolution this format allowed is 0.1°C .
- 4917 A DeltaTemperature of 0xFF indicates that the temperature offset is not used.
- 4918 If a temperature offset is sent that causes the heating or cooling temperature set point to exceed
4919 the limit boundaries that are programmed into the thermostat, the thermostat should respond by
4920 setting the temperature at the limit.
- 4921 **Cooling Temperature Set Point (optional):** Requested cooling set point in 0.01 degrees
4922 Celsius.
- 4923 **Heating Temperature Set Point (optional):** Requested heating set point in 0.01 degrees
4924 Celsius.
- 4925 Cooling and heating temperature set points will be defined and calculated per the
4926 *LocalTemperature* attribute in the Thermostat Cluster [B1].
- 4927 These fields represent the temperature in degrees Celsius, as follows:
- 4928 Cooling Temperature Set Point / 100 = temperature in degrees Celsius
4929 Where $-273.15^{\circ}\text{C} \leq$ temperature $\leq 327.67^{\circ}\text{C}$, corresponding to a Cooling and/or Heating
4930 Temperature Set Point in the range 0x954d to 0x7fff.
- 4931 The maximum resolution this format allows is 0.01°C .
- 4932 A Cooling or Heating Temperature Set Point of 0x8000 indicates that the temperature set
4933 point is not used.
- 4934 If a temperature is sent that exceeds the temperature limit boundaries that are programmed
4935 into the thermostat, the thermostat should respond by setting the temperature at the limit.
- 4936 The thermostat shall not use a Cooling or Heating Temperature Set Point that causes the
4937 device to use more energy than the normal setting.
- 4938 When both a Temperature Offset and a Temperature Set Point are provided, the thermostat may
4939 use either as defined by the device manufacturer. The thermostat should use the setting that
4940 provides the lowest energy consumption.
- 4941 **Average Load Adjustment Percentage (optional):** Defines a maximum energy usage limit as
4942 a percentage of the client implementations specific average energy usage. The load adjustment

percentage is added to 100% creating a percentage limit applied to the client implementations specific average energy usage. A -10% load adjustment percentage will establish an energy usage limit equal to 90% of the client implementations specific average energy usage. Each load adjustment percentage is referenced to the client implementations specific average energy usage. There are no cumulative effects.

The range of this field is -100 to +100 with a resolution of 1 percent. A -100% value equals a total load shed. A 0% value will limit the energy usage to the client implementation's specific average energy usage. A +100% value will limit the energy usage to double the client implementation's specific average energy usage.

A value of 0x80 indicates the field is not used. All other values are reserved for future use.

Duty Cycle (optional): Defines the maximum On state duty cycle as a percentage of time. Example, if the value is 80, the device would be in an “on state” for 80% of the time for the duration of the event. Range of the value is 0 to 100. A value of 0xFF indicates the field is not used. All other values are reserved for future use.

Duty cycle control is a device specific issue and shall be managed by the device manufacturer. It is expected that the duty cycle of the device under control will span the shortest practical time period in accordance with the nature of the device under control and the intent of the request for demand reduction. For typical Device Classes, three minutes⁷ for each 10% of duty cycle is recommended. It is expected that the “off state” will precede the “on state”.

Event Control (mandatory): Identifies additional control options for the event. The BitMap for this field is described in Table D-4.

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

Note: The randomization attribute will be used in combination with two bits to determine if the Event Start and Stop Times are randomized. By default devices will randomize the start 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 these values.

4970 **D.2.2.3.1.1.2 When Generated**

This command is generated when the ESI wants to control one or more load control devices, usually as the result of an energy curtailment command from the Smart Energy network.

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.

5001 If the Device Class and/or Utility Enrollment Group fields don't apply to your End Device, the
5002 *Cancel Load Control Event* command is ignored.

5003 Device Class and/or Utility Group fields must be the same for a *Cancel Load Control Event*
5004 command as they were for the command to create the event. Should these fields be different
5005 there is no defined behavior for how DRLC servers should maintain their tables for replying to
5006 *Get Scheduled Events* commands.

5007 **Cancel Control (mandatory):** The encoding of the Cancel Control is listed in Table D-5.

5008 **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 <i>Cancel Load Control Event</i> 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

5009
5010 **Effective Time (mandatory):** UTC Timestamp representing when the canceling of the event is
5011 scheduled to start. An effective time of 0x00000000 is a special time denoting “now.” If the
5012 device would send an event with an Effective Time of now, adjust the Duration In Minutes field
5013 to correspond to the remainder of the event.

5014 **Note: This field is deprecated; a *Cancel Load Control* command shall now take immediate
5015 effect. A value of 0x00000000 shall be used in all *Cancel Load Control* commands**

5016 **D.2.2.3.2.1.2 When Generated**

5017 This command is generated when the ESI wants to cancel previously scheduled control of one or
5018 more load control devices, usually as the result of an energy curtailment command from the
5019 Smart Energy network.

5020 **D.2.2.3.2.1.3 Responses to *Cancel Load Control Event***

5021 The server receives the cluster-specific commands detailed in sub-clause D.2.3.3.1.

5022 **Note: If the *Cancel Load Control* Event command is received after the event has ended, the
5023 device shall reply using the “Report Event Status Command” with an Event Status of “Rejected
5024 - Invalid Cancel Command (Undefined Event)”.**

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 **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.

5045 **D.2.3 Client**

5046 This section identifies the attributes and commands provided by Client devices.

5047 **D.2.3.1 Dependencies**

5048 Devices receiving and acting upon *Load Control Event* commands must be capable of storing
5049 and supporting at least three unique instances of events. As a highly recommended recovery
5050 mechanism, when maximum storage of events has been reached and additional Load Control
5051 Events are received that are unique (not superseding currently stored events), devices should
5052 ignore additional Load Control Events and when storage becomes available, utilize the
5053 *GetScheduledEvents* command to retrieve any previously ignored events.

5054 Events carried using this cluster include a timestamp with the assumption that target devices
5055 maintain a real time clock. Devices can acquire and synchronize their internal clocks with the
5056 ESI as described in sub-clause 5.12.1.1.

5057 Devices MAY ‘drop’ events received before they have received and resolved time (‘dropping’ an
5058 event is defined as sending a default response with status code SUCCESS).

5059 If a device does not support a real time clock, it's assumed the device will ignore all values
5060 within the Time field except the “Start Now” value.

5061 Additionally, for devices without a real time clock it's assumed those devices will utilize a
5062 method (i.e. ticks, countdowns, etc.) to approximate the correct duration period.

5063 **D.2.3.2 Client Cluster Attributes**

5064 **Table D-7– Demand Response Client Cluster Attributes**

Identifier	Name	Type	Range	Access	Default	Mandatory/ Optional
0x0000	<i>UtilityEnrollment Group</i>	Unsigned 8-bit Integer	0x00 to 0xFF	Read/ Write	0x00	M
0x0001	<i>StartRandomizeMinutes</i>	Unsigned 8-bit Integer	0x00 to 0x3C	Read/ Write	0x1E	M
0x0002	<i>StopRandomizeMinutes</i>	Unsigned 8-bit Integer	0x00 to 0x3C	Read/ Write	0x1E	M
0x0003	<i>DeviceClassValue</i>	Unsigned 16-bit Integer	0x0000 to 0xFFFF	Read/ Write	-	M
0x0004 to 0xFFFF	Reserved					

5065

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.

5101 In-Home Displays should report the device classes that they are interested in. An IHD that
 5102 wishes to display all possible Load Control Events, even for classes not yet defined, should
 5103 indicate a device class of 0xFFFF; this will allow DRLC servers to optimize the number of
 5104 DRLC events they unicast, such that they are only sent to those devices that are interested in
 5105 them.

5106 **D.2.3.3 Commands Generated**

5107 The command IDs generated by the Demand Response and Load Control client cluster are
 5108 listed in Table D-8.

5109

5110 **Table D-8– Generated Command IDs for the Demand Response and Load Control Client**

Command Identifier Field Value	Description	Mandatory/ Optional
0x00	<i>Report Event Status</i>	M
0x01	<i>Get Scheduled Events</i>	M
0x02 – 0xff	Reserved	

5111

5112 **D.2.3.3.1 Report Event Status Command**

5113 **D.2.3.3.1.1 Payload Format**

5114 The *Report Event Status* command payload shall be formatted as illustrated in Figure D-5.

5115

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)

5116

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)

Field Name	Average Load Adjustment Percentage Applied (O)	Duty Cycle Applied (O)	Event Control (M)	Signature Type (M)	Signature (O)

5117

Figure D-5– Report Event Status Command Payload5118 **D.2.3.3.1.1.1 Payload Details**

5119 **Issuer Event ID (mandatory):** Unique identifier generated by the Energy provider. The value of
 5120 this field allows matching of Event reports with a specific Demand Response and Load Control
 5121 event. It's expected the value contained in this field is a unique number managed by upstream
 5122 systems or a UTC based time stamp (UTCTime data type) identifying when the Load Control
 5123 Event was issued.

5124 **Event Status (mandatory):** Table D-9 lists the valid values returned in the Event Status field.5125 **Table D-9– Event Status Field Values**

Value	Description
0x00	Reserved for future use.
0x01	<i>Load Control Event</i> 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	<i>Load Control Event</i> command Rejected
0xFF	Reserved for future use.

- 5126
5127 Should a device issue one or more “OptOut” or “OptIn” RES commands during an event that
5128 is eventually cancelled, the event shall be recorded as a cancelled event (Status = 0x06) at its
5129 effective time.
- 5130 Should a device issue one or more “OptOut” or “OptIn” RES commands during an event that
5131 is not cancelled, the event shall be recorded as partially completed based on the last RES
5132 command sent (Status = 0x08 or 0x09).
- 5133 When a device returns a status of 0xFD (Rejected - Invalid Cancel Command (Undefined
5134 Event)), all optional fields should report their “Ignore” values.
- 5135 When a device receives a duplicate RES command, it should ignore the duplicate commands.
5136 Please note: As a recommended best practice, ESI applications should provide a mechanism to
5137 assist in filtering duplicate messages received on the WAN.
- 5138 **Event Status Time (mandatory):** UTC Timestamp representing when the event status
5139 occurred. This field shall not use the value of 0x00000000.
- 5140 **Criticality Level Applied (mandatory):** Criticality Level value applied by the device, see the
5141 corresponding field in the *Load Control Event* command for more information.
- 5142 **Cooling Temperature Set Point Applied (optional):** Cooling Temperature Set Point value
5143 applied by the device, see the corresponding field in the *Load Control Event* command for more
5144 information. The value 0x8000 means that this field has not been used by the end device.
- 5145 **Heating Temperature Set Point Applied (optional):** Heating Temperature Set Point value
5146 applied by the device, see the corresponding field in the *Load Control Event* command for more
5147 information. The value 0x8000 means that this field has not been used by the end device.
- 5148 **Average Load Adjustment Percentage Applied (optional):** Average Load Adjustment
5149 Percentage value applied by the device, see the corresponding field in the *Load Control Event*
5150 command for more information. The value 0x80 means that this field has not been used by the
5151 end device.
- 5152 **Duty Cycle Applied (optional):** Defines the maximum On state duty cycle applied by the
5153 device. The value 0xFF means that this field has not been used by the end device. Refer to sub-
5154 clause D.2.2.3.1.1.1.
- 5155 **Event Control (mandatory):** Identifies additional control options for the event. Refer to sub-
5156 clause D.2.2.3.1.1.1.
- 5157 **Signature Type (mandatory):** An 8-bit Unsigned integer enumerating the type of algorithm
5158 use to create the Signature. The enumerated values are:

Enumerated Value	Signature Type
0x00	No Signature
0x01	ECDSA
0x02 to 0xFF	Reserved

5159
5160 If the signature field is not used, the signature type shall be set to 0x00, which will be used to
5161 indicate “no signature”. The signature field shall be filled with (48) 0xFF values.

5162 **Signature (optional):** A non-repudiation signature created by using the Matyas-Meyer-Oseas
5163 hash function (specified in Annex B.6 in [B3]) used in conjunction with ECDSA. The
5164 signature creation process will occur in two steps:

5165 1 Pass the first ten fields, which includes all fields up to the Signature field, of the *Report Event*
5166 *Status* command (listed in Figure D-5) through ECDSA using the device's ECC Private Key,
5167 generating the signature (r,s) .

5168 *Note: ECDSA internally uses the MMO hash function in place of the internal SHA-1 hash*
5169 *function.*

5170 2 Concatenate ECDSA signature components (r,s) and place into the Signature field within the
5171 *Report Event Status* command.

5172 *Note: the lengths of r and s are implicit, based on the curve used. Verifying the signature will require*
5173 *breaking the signature field back into the discrete components r and s, based on the length.*

5174 D.2.3.3.1.2 When Generated

5175 This command is generated when the client device detects a change of state for an active Load
5176 Control event. (The transmission of this command should be delayed after a random delay
5177 between 0 and 5 seconds, to avoid a potential storm of packets.)

5178 D.2.3.3.2 Get Scheduled Events Command

5179 *Note: The handling of this command is currently under review, and is likely to change in the next*
5180 *revision of the specification. Refer to CCB 1297 (and associated document 12-0180-00) for*
5181 *further information*

5182 This command is used to request that all scheduled Load Control Events, starting at or after the
5183 supplied Start Time, are re-issued to the requesting device. When received by the Server, one or more
5184 *Load Control Event* commands (see sub-clause D.2.2.3.1) will be sent covering both active and
5185 scheduled Load Control Events.

5186 D.2.3.3.2.1 Payload Format

5187 The *Get Scheduled Events* command payload shall be formatted as illustrated in Figure D-6

Octets	4	1
Data Type	UTCTime	Unsigned 8-bit integer
Field Name	Start Time (M)	Number of Events (M)

5188

Figure D-6– Get Scheduled Events Command Payload

5189 **Start Time (mandatory):** UTC Timestamp representing the minimum ending time for any
5190 scheduled or currently active events to be resent. If either command has a Start Time of
5191 0x00000000, replace that Start Time with the current time stamp.

5192 **Number of Events (mandatory):** Represents the maximum number of events to be sent. A
5193 value of 0 would indicate all available events are to be returned. Example: Number of
5194 Events = 1 would return the first event with an EndTime greater than or equal to the value of
5195 Start Time field in the *Get Scheduled Events* command (EndTime would be StartTime plus
5196 Duration of the event listed in the device's event table).

5197 **D.2.3.3.2.2 When Generated**

5198 This command is generated when the client device wishes to verify the available Load Control
5199 Events or after a loss of power/reset occurs and the client device needs to recover currently
5200 active or scheduled Load Control Events.

5201 A ZCL Default Response with status NOT_FOUND shall be returned when there are no events
5202 available.

5203 **D.2.3.4 Commands Received**

5204 The client receives the cluster-specific commands detailed in sub-clause D.2.2.

5205 **D.2.3.5 Attribute Reporting**

5206 Attribute reporting is not expected to be used for this cluster. The Client side attributes are
5207 not expected to be changed by the Client, only used during Client operations.

5208 **D.2.4 Application Guidelines**

5209 The criticality level is sent by the utility to the load control device to indicate how much load
5210 reduction is requested. The utility is not required to use all of the criticality levels that are
5211 described in this specification. A load control device is not required to provide a unique
5212 response to each criticality level that it may receive.

5213 The Average Load Adjustment Percentage, temperature offsets, and temperature set points are
5214 used by load control devices and energy management systems on a “voluntary” or “optional”
5215 basis. These devices are not required to use the values that are provided by the utility. They are
5216 provided as a recommendation by the utility.

5217 The load control device shall, in a manner that is consistent with this specification, accurately
5218 report event participation by way of the Report Event Status message.

5219 The Average Load Adjustment Percentage is sent by the utility to the load control device to
5220 indicate how much load reduction is requested. The load control device may respond to this
5221 information in a unique manner as defined by the device manufacturer.

5222 The Duty Cycle is sent by the utility to the load control device to indicate the maximum “On
5223 state” for a device. The control device may respond to this information in a unique manner as
5224 defined by the device manufacturer.

5225 The cooling temperature offset may be sent by the utility to the load shed control to indicate how
5226 much indoor cooling temperature offset is requested. Response of a load control device to this
5227 information is not mandatory. The control device may respond to this information in a
5228 unique manner as defined by the device manufacturer.

5229 The heating temperature offset may be sent by the utility to the load control device to indicate how
5230 much indoor heating temperature offset is requested. The control device may respond to this
5231 information in a unique manner as defined by the device manufacturer.

5232 The cooling temperature may be sent by the utility to the load control device to indicate the
5233 indoor cooling temperature setting that is requested. The control device may respond to this
5234 information in a unique manner as defined by the device manufacturer.

5235 The heating temperature may be sent by the utility to the load control device to indicate the
5236 indoor heating temperature setting that is requested. The control device may respond to this
5237 information in a unique manner as defined by the device manufacturer.

5238 *Note: The most recent Load Control Event supersedes any previous Load Control Event
5239 command for the set of Device Classes and groups for a given time. Nested events and
5240 overlapping events are not allowed. The current active event will be terminated if a new event
5241 is started.*

5242 [D.2.4.1 Load Control Rules, Server](#)

5243 [**D.2.4.1.1 Load Control Server, Identifying Use of SetPoint and Offset Fields**](#)

5244 The use of the fields, Heating and Cooling Temperature Set Points and Heating and Cooling
5245 Temperature Offsets is optional. All fields in the payload must be populated. Non-use of
5246 these fields by the Server is indicated by using the following values: 0x8000 for Set Points
5247 and 0xFF for Offsets. When any of these four fields are indicated as optional, they shall be
5248 ignored by the client.

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

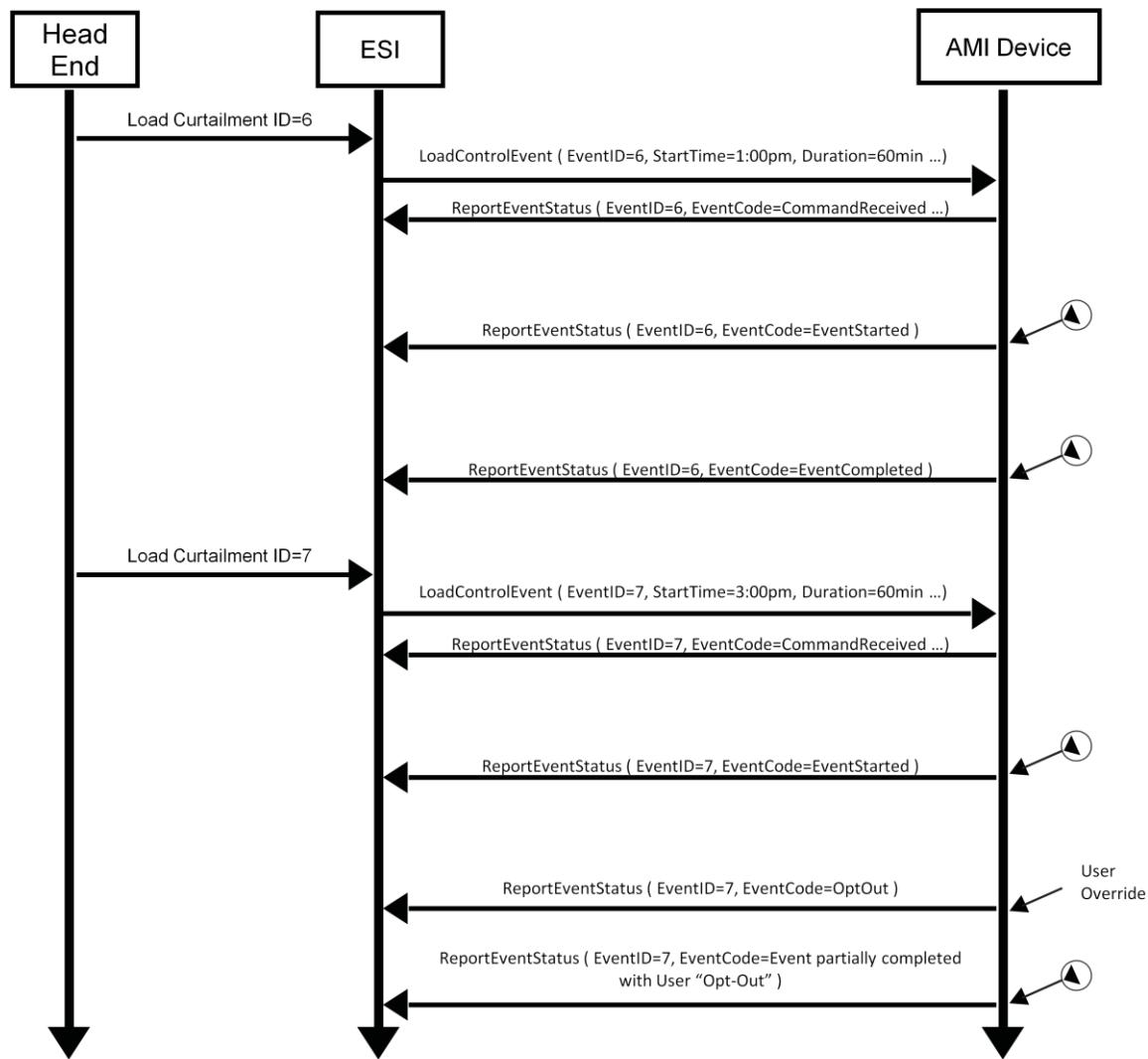
5282 HVAC control must obey the event rules that would have been enforced if the system had been
5283 operating in that mode at the start of the active event.

5284 An event override message, “opt-out”, will be sent by the load control device or energy
5285 management system if the operator chooses not to participate in a demand response event by
5286 taking action to override the programmed demand reduction response. The override message
5287 will be sent at the start of the event. In the case where the event has been acknowledged and
5288 started, the override message will be sent when the override occurs.

5289 **D.2.4.2.6Demand Response and Load Control Transaction Examples**

5290 The following example in Figure D-7 depicts the transactions that would take place for two events,
5291 one that is successful and another that is overridden by the user.

5292

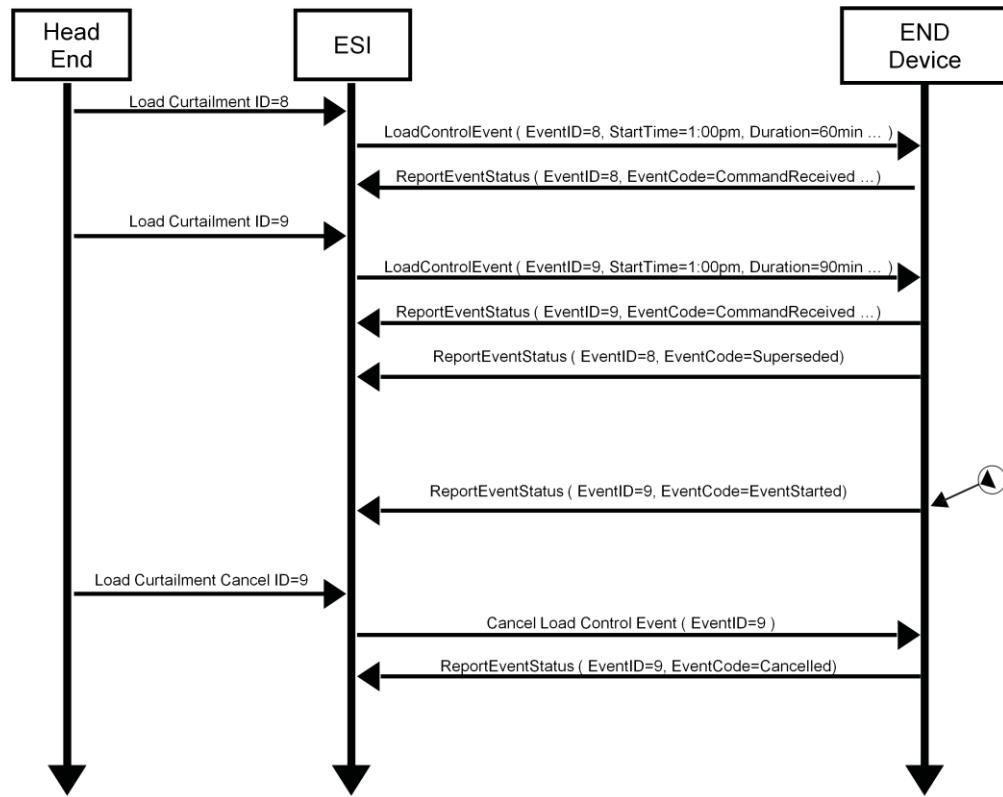


5293
5294

5295
5296
5297

Figure D-7– Example of Both a Successful and an Overridden Load Curtailment Event

5298 The example in Figure D-8 depicts the transactions that would take place when an event is
 5299 superseded by an event that is eventually cancelled.



5300

Figure D-8— Example of a Load Curtailment Superseded and Another Cancelled

5301 Please refer to Annex E for more information regarding the management and behavior of
 5302 overlapping events.
 5303

5306

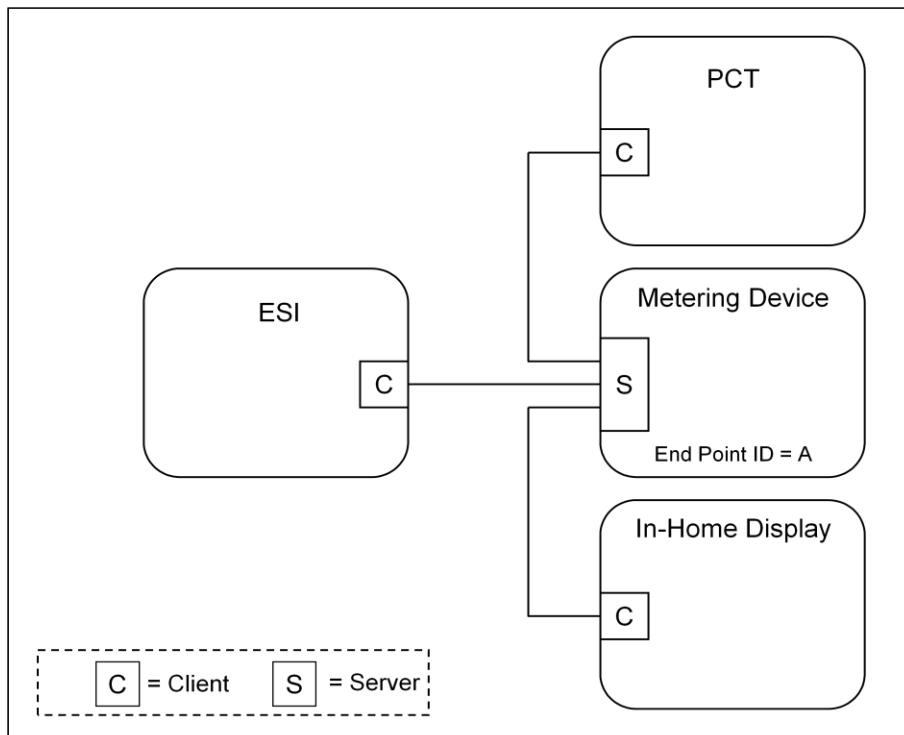
5307

5308 **D.3 Metering Cluster**

5309 **D.3.1 Overview**

5310 The Metering Cluster provides a mechanism to retrieve usage information from Electric, Gas,
 5311 Water, and potentially Thermal metering devices. These devices can operate on either battery or
 5312 mains power, and can have a wide variety of sophistication. The Metering Cluster is designed to
 5313 provide flexibility while limiting capabilities to a set number of metered information types. More
 5314 advanced forms or data sets from metering devices will be supported in the Smart Energy
 5315 Tunneling Cluster, which will be defined in sub-clause D.6.

5316 The following figures identify three configurations as examples utilizing the Metering
 5317 Cluster.

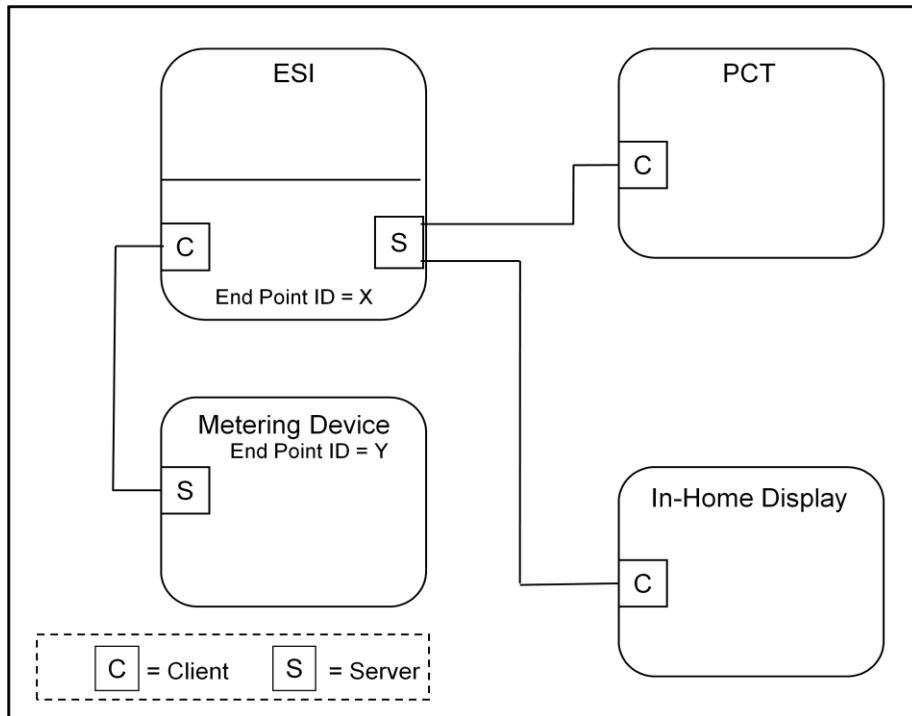


5318
 5319 **Figure D-9– Standalone ESI Model with Mains Powered Metering Device**

5320

5321 In the example shown in Figure D-9, the metering device is the source of information
 5322 provided via the Metering Cluster Server.

5323



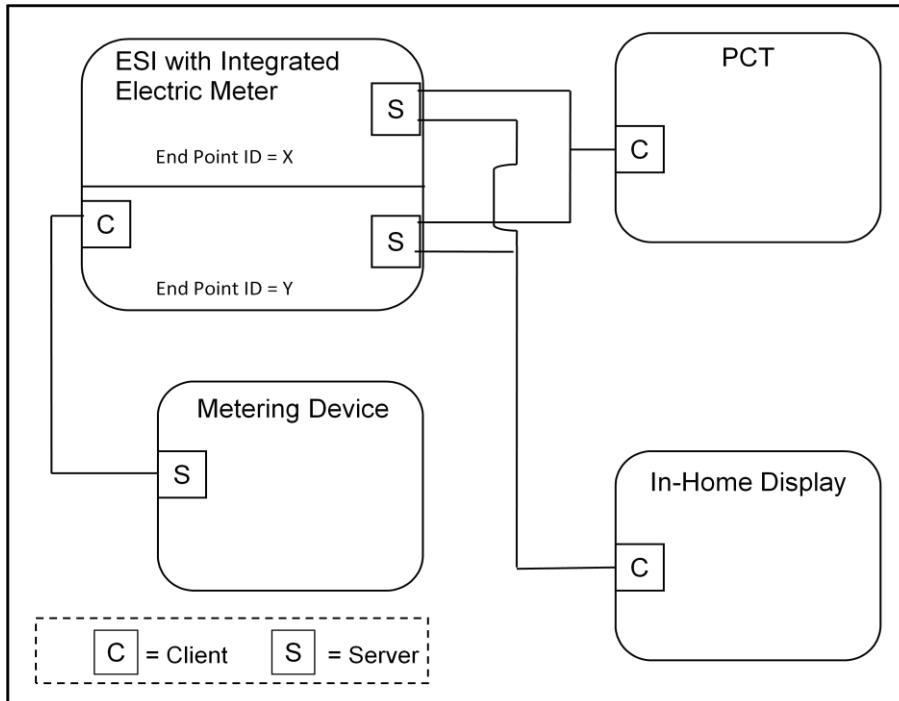
5324
5325

Figure D-10– Standalone ESI Model with Battery Powered Metering Device

5326
5327
5328 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).

5329
5330

5331
5332
5333
5334
5335
5336



5337
5338

Figure D-11– ESI Model with Integrated Metering Device

5339
5340
5341
5342
5343
5344
5345
5346
5347
5348
5349
5350

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).

5351
5352
5353
5354
5355
5356
5357

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.

5358
5359
5360

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).

5361 **D.3.2 Server**5362 **D.3.2.1 Dependencies**

5363 Subscribed reporting of Metering attributes.

5364 **D.3.2.2 Attributes**

5365 For convenience, the attributes defined in this specification are arranged into sets of related
 5366 attributes; each set can contain up to 256 attributes. Attribute identifiers are encoded such that the
 5367 most significant Octet specifies the attribute set and the least significant Octet specifies the
 5368 attribute within the set. The currently defined attribute sets are listed in Table D-10.

5369 *Note: Certain attributes within this cluster are provisional and not certifiable. Refer to the*
 5370 *individual attribute sets for details of the relevant attributes.*

5371 **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

5372

5373 **D.3.2.2.1 Reading Information Set**

5374 The following set of attributes provides a remote access to the reading of the Electric, Gas, or
 5375 Water metering device. A reading must support at least one register which is the actual total
 5376 summation of the delivered quantity (kWh, m³, ft³, ccf, US gl).

5377 Please note: In the following attributes, the term “Delivered” refers to the quantity of Energy,
 5378 Gas, or Water that was delivered to the customer from the utility. Likewise, the term

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	<i>CurrentSummationDelivered</i>	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	M
0x0001	<i>CurrentSummationReceived</i>	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0002	<i>CurrentMaxDemandDelivered</i>	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0003	<i>CurrentMaxDemandReceived</i>	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0004	<i>DFTSummation</i>	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0005	<i>Daily Freeze Time</i>	Unsigned 16-bit Integer	0x0000 to 0x173B	Read Only	0x0000	O
0x0006	<i>PowerFactor</i>	Signed 8-bit Integer	-100 to +100	Read Only	0x00	O
0x0007	<i>ReadingSnapshotTime</i>	UTCTime		Read Only	-	O
0x0008	<i>CurrentMaxDemandDeliveredTime</i>	UTCTime		Read Only	-	O
0x0009	<i>CurrentMaxDemandReceivedTime</i>	UTCTime		Read Only	-	O
0x000A	<i>DefaultUpdatePeriod</i>	Unsigned 8-bit Integer	0x00 to 0xFF	Read Only	0x1E	O
0x000B	<i>FastPollUpdatePeriod</i>	Unsigned 8-bit Integer	0x00 to 0xFF	Read Only	0x05	O
0x000C	<i>CurrentBlockPeriodConsumptionDelivered</i>	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O

0x000D	<i>DailyConsumption Target</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x000E	<i>CurrentBlock</i>	8-bit Enumeration	0x00 to 0x10	Read Only	-	O
0x000F	<i>ProfileInterval Period</i>	8-bit Enumeration	0x00 to 0xFF	Read Only	-	O
0x0010	<i>Deprecated</i> ³					
0x0011	<i>PresetReading Time</i>	Unsigned 16-bit Integer	0x0000 to 0x173B	Read Only	0x0000	O
0x0012	<i>VolumePerReport</i>	Unsigned 16-bit Integer	0x0000 to 0xFFFF	Read Only	-	O
0x0013	<i>FlowRestriction</i>	Unsigned 8-bit Integer	0x00 to 0xFF	Read Only	-	O
0x0014	<i>Supply Status</i>	8-bit Enumeration	0x00 to 0xFF	Read Only	-	O
0x0015	<i>CurrentInletEnergy CarrierSummation</i>	Unsigned 48-bit Integer	0x0000000000000000 to 0xFFFFFFFFFFFFFF	Read Only	-	O ⁴
0x0016	<i>CurrentOutletEnergy Carrier Summation</i>	Unsigned 48-bit Integer	0x0000000000000000 to 0xFFFFFFFFFFFFFF	Read Only	-	O
0x0017	<i>InletTemperature</i>	Signed 24-bit Integer	-8,388,607 to 8,388,607	Read Only	-	O ⁵
0x0018	<i>OutletTempera ture</i>	Signed 24-bit Integer	-8,388,607 to 8,388,607	Read Only	-	O ⁶
0x0019	<i>ControlTemperature</i>	Signed 24-bit Integer	-8,388,607 to 8,388,607	Read Only	-	O

³ CCB 1886⁴ CCB 1999⁵ CCB 1999⁶ CCB 1999

0x001A	<i>CurrentInletEnergy CarrierDemand</i>	Signed 24-bit Integer	-8,388,607 to 8,388,607	Read Only	-	O
0x001B	<i>CurrentOutletEnergy Carrier Demand</i>	Signed 24-bit Integer	-8,388,607 to 8,388,607	Read Only	-	O
0x001C	<i>PreviousBlockPeriod ConsumptionDelivered</i>	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					

5385

5386 D.3.2.2.1.1 CurrentSummationDelivered Attribute

5387 *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.

5391 D.3.2.2.1.2 CurrentSummationReceived Attribute

5392 *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.

5395 D.3.2.2.1.3 CurrentMaxDemandDelivered Attribute

5396 *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.

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

5430 *InstantaneousDemand* must be updated at the *DefaultUpdatePeriod*. The *DefaultUpdatePeriod*
5431 may apply to other attributes as defined by the device manufacturer.

5432 D.3.2.2.1.12 FastPollUpdatePeriod Attribute

5433 The *FastPollUpdatePeriod* attribute represents the interval (seconds) at which the
5434 *InstantaneousDemand* attribute is updated when in fast poll mode. *InstantaneousDemand* may be
5435 continuously updated as new measurements are acquired, but at a minimum,
5436 *InstantaneousDemand* must be updated at the *FastPollUpdatePeriod*. The
5437 *FastPollUpdatePeriod* may apply to other attributes as defined by the device manufacturer.

5438 D.3.2.2.1.13 CurrentBlockPeriodConsumptionDelivered Attribute

5439 The *CurrentBlockPeriodConsumptionDelivered* attribute represents the most recent summed
5440 value of Energy, Gas or Water delivered and consumed in the premises during the Block Tariff
5441 Period.

5442 The *CurrentBlockPeriodConsumptionDelivered* is reset at the start of each Block Tariff Period.

5443 D.3.2.2.1.14 DailyConsumptionTarget Attribute

5444 The *DailyConsumptionTarget* attribute is a daily target consumption amount that can be
5445 displayed to the consumer on a HAN device, with the intent that it can be used to compare to
5446 actual daily consumption (e.g. compare to the *CurrentDayConsumptionDelivered*).

5447 This may be sent from the utility to the ESI, or it may be derived. Although intended to be
5448 based on Block Thresholds, it can be used for other targets not related to blocks. The
5449 formatting will be based on the *HistoricalConsumptionFormatting* attribute.

5450 Example: If based on a Block Threshold, the *DailyConsumptionTarget* could be calculated
5451 based on the number of days specified in the Block Tariff Period and a given Block Threshold as
5452 follows: $\text{DailyConsumptionTarget} = \text{BlockNThreshold} / ((\text{BlockPeriodDuration} / 60) / 24)$.
5453 Example: If the target is based on a *Block1Threshold* of 675kWh and where 43200
5454 *BlockThresholdPeriod* is the number of minutes in the billing period (30 days), the
5455 *ConsumptionDailyTarget* would be $675 / ((43200 / 60) / 24) = 22.5$ kWh per day.

5456 D.3.2.2.1.15 CurrentBlock Attribute

5457 When Block Tariffs are enabled, *CurrentBlock* is an 8-bit Enumeration which indicates the
5458 currently active block. If blocks are active then the current active block is based on the
5459 *CurrentBlockPeriodConsumptionDelivered* and the block thresholds. Block 1 is active when the⁷
5460 value of *CurrentBlockPeriodConsumptionDelivered* is less than or equal to the⁷
5461 *Block1Threshold* value, Block 2 is active when *CurrentBlockPeriodConsumptionDelivered* is
5462 greater than *Block1Threshold* value and less than or equal to the⁸ *Block2Threshold* value, and so

⁷ CCB 1679

⁸ CCB 1679

5463 on. Block 16 is active when the value of *CurrentBlockPeriodConsumptionDelivered* is greater
 5464 than *Block15Threshold* value.

5465

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

5466 D.3.2.2.1.16 ProfileIntervalPeriod Attribute

5467 The *ProfileIntervalPeriod* attribute is currently included in the *Get Profile Response* command
 5468 payload, but does not appear in an attribute set. This represents the duration of each interval.
 5469 *ProfileIntervalPeriod* represents the interval or time frame used to capture metered Energy, Gas, and
 5470 Water consumption for profiling purposes. The enumeration for this field shall match one of the
 5471 *ProfileIntervalPeriod* values defined in sub-clause D.3.2.3.1.1.⁹

5472 D.3.2.2.1.17 PresetReadingTime

5473 The *PresetReadingTime* attribute represents the time of day (in quarter hour increments) at
 5474 which the meter will wake up and report a register reading even if there has been no
 5475 consumption for the previous 24 hours. *PresetReadingTime* is an unsigned 16-bit value
 5476 representing the hour and minutes. The byte usages are:

⁹ CCB 1886

5477 **Bits 0 to 7:** Range of 0 to 0x3B representing the number of minutes past the top of the hour.

5478 **Bits 8 to 15:** Range of 0 to 0x17 representing the hour of the day (in 24-hour format).

5479 E.g.: A setting of 0x172D would represent 23:45 hours or 11:45 pm; a setting of 0x071E would
5480 represent 07:30 hours or 7:30 am. A setting of 0xFFFF indicates this feature is disabled. The use
5481 of Attribute Reporting Configuration is optional.

5482 D.3.2.2.1.18 VolumePerReport Attribute

5483 The *VolumePerReport* attribute represents the volume per report increment from the water or
5484 gas meter. For example a gas meter might be set to report its register reading for every time 1
5485 cubic meter of gas is used. For a water meter it might report the register value every 10 liters of
5486 water usage.

5487 D.3.2.2.1.19 FlowRestriction Attribute

5488 The *FlowRestriction* attribute represents the volume per minute limit set in the flow restrictor.
5489 This applies to water but not for gas. A setting of 0xFF indicates this feature is disabled.

5490 D.3.2.2.1.20 SupplyStatus Attribute

5491 The *SupplyStatus* attribute represents the state of the supply at the customer's premises. The
5492 enumerated values for this field are outlined in Table D-13:

5493 **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

5494

5495 D.3.2.2.1.21 CurrentInletEnergyCarrierSummation Attribute

5496 *CurrentInletEnergyCarrierSummation* is the current integrated volume of a given energy carrier
5497 measured on the inlet. The formatting and unit of measure for this value is specified in the
5498 *EnergyCarrierUnitOfMeasure* and *EnergyCarrierSummationFormatting* attributes (refer to Table
5499 D-25).

5500 The Energy consumption registered in *CurrentSummationDelivered* is not necessarily a direct
5501 function of this value. The quality of the energy carrier may vary from day to day, e.g. Gas may
5502 have different quality.

5503 For heat and cooling meters the energy carrier is water at high or low temperature, the energy
5504 withdrawn from such a system is a function of the flow and the inlet and outlet temperature.

5505 D.3.2.2.1.22 CurrentOutletEnergyCarrierSummation Attribute

5506 *CurrentOutletEnergyCarrierSummation* is the current integrated volume of a given energy
5507 carrier measured on the outlet. The formatting and unit of measure for this value is specified in
5508 the *EnergyCarrierUnitOfMeasure* and *EnergyCarrierSummationFormatting* attributes (refer to
5509 Table D-25).

5510 D.3.2.2.1.23 InletTemperature Attribute

5511 *InletTemperature* is the temperature measured on the energy carrier inlet.

5512 The formatting and unit of measure for this value is specified in the *TemperatureUnitOfMeasure*
5513 and *TemperatureFormatting* attributes (refer to Table D-25).

5514 D.3.2.2.1.24 OutletTemperature Attribute

5515 *OutletTemperature* is the temperature measured on the energy carrier outlet.

5516 The formatting and unit of measure for this value is specified in the *TemperatureUnitOfMeasure*
5517 and *TemperatureFormatting* attributes (refer to Table D-25).

5518 D.3.2.2.1.25 ControlTemperature Attribute

5519 *ControlTemperature* is a reference temperature measured on the meter used to validate the
5520 Inlet/Outlet temperatures.

5521 The formatting and unit of measure for this value is specified in the
5522 *TemperatureUnitOfMeasure* and *TemperatureFormatting* attributes (refer to Table D-25).

5523 D.3.2.2.1.26 CurrentInletEnergyCarrierDemand Attribute

5524 *CurrentInletEnergyCarrierDemand* is the current absolute demand on the energy carrier inlet.

5525 The formatting and unit of measure for this value is specified in the
5526 *EnergyCarrierUnitOfMeasure* and *EnergyCarrierDemandFormatting* attributes (refer to Table
5527 D-25).

5528 For a heat or cooling meter this will be the current absolute flow rate measured on the inlet.

5529 D.3.2.2.1.27 CurrentOutletEnergyCarrierDemand Attribute

5530 *CurrentOutletEnergyCarrierDemand* is the current absolute demand on the energy carrier
5531 outlet.

5532 The formatting and unit of measure for this value is specified in the
5533 *EnergyCarrierUnitOfMeasure* and *EnergyCarrierDemandFormatting* attributes (refer to Table
5534 D-25).

5535 For a heat or cooling meter this will be the current absolute flow rate measured on the outlet.

5536 D.3.2.2.1.28 PreviousBlockPeriodConsumptionDelivered Attribute

5537 The *PreviousBlockPeriodConsumptionDelivered* attribute represents the total value of Energy,
5538 Gas or Water delivered and consumed in the premises at the end of the previous Block Tariff
5539 Period. If supported, the *PreviousBlockPeriodConsumptionDelivered* attribute is updated at the
5540 end of each Block Tariff Period.

5541 D.3.2.2.1.29 CurrentBlockPeriodConsumptionReceived Attribute

5542 The *CurrentBlockPeriodConsumptionReceived* attribute represents the most recent summed
5543 value of Energy, Gas or Water received by the energy supplier from the premises during the
5544 Block Tariff Period. The *CurrentBlockPeriodConsumptionReceived* attribute is reset at the start
5545 of each Block Tariff Period.

5546 D.3.2.2.1.30 CurrentBlockReceived Attribute

5547 When Block Tariffs are enabled, *CurrentBlockReceived* is an 8-bit Enumeration which indicates
5548 the currently active block. If blocks are active then the current active block is based on the
5549 *CurrentBlockPeriodConsumptionReceived* and the block thresholds. Block 1 is active when the
5550 value of *CurrentBlockPeriodConsumptionReceived* is less than or equal to the Block1Threshold
5551 value, Block 2 is active when *CurrentBlockPeriodConsumptionReceived* is greater than
5552 Block1Threshold value and less than or equal to the Block2Threshold value, and so on. Block 16
5553 is active when the value of *CurrentBlockPeriodConsumptionReceived* is greater than
5554 Block15Threshold value. Refer to Table D-12 for block enumerations.

5555 DFTSummationReceived Attribute

5556 *DFTSummationReceived* represents a snapshot of attribute *CurrentSummationReceived* captured
5557 at the time indicated by the *DailyFreezeTime* attribute (see D.3.2.2.1.6).

5558 If optionally provided, *DFTSummationReceived* is updated once every 24 hours and captured at
5559 the time set in the *DailyFreezeTime* attribute (see D.3.2.2.1.6).

5560 D.3.2.2.1.32 ActiveRegisterTierDelivered Attribute

5561 The *ActiveRegisterTierDelivered* attribute indicates the current register tier that the energy
5562 consumed is being accumulated against. Valid values for this attribute are defined in Table D-98.

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.3.2.2.2Summation 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	<i>CurrentTier1 SummationDelivered</i>	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0101	<i>CurrentTier1 SummationReceived</i>	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0102	<i>CurrentTier2 SummationDelivered</i>	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0103	<i>CurrentTier2 SummationReceived</i>	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0104	<i>CurrentTier3 SummationDelivered</i>	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O

0x0105	<i>CurrentTier3 SummationReceived</i>	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0106	<i>CurrentTier4 SummationDelivered</i>	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0107	<i>CurrentTier4 SummationReceived</i>	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0108	<i>CurrentTier5 SummationDelivered</i>	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0109	<i>CurrentTier5 SummationReceived</i>	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x010A	<i>CurrentTier6 SummationDelivered</i>	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x010B	<i>CurrentTier6 SummationReceived</i>	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x010C	<i>CurrentTier7 SummationDelivered</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x010D	<i>CurrentTier7 SummationReceived</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x010E	<i>CurrentTier8 SummationDelivered</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x010F	<i>CurrentTier8 SummationReceived</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0110	<i>CurrentTier9 SummationDelivered</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0111	<i>CurrentTier9 SummationReceived</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0112	<i>CurrentTier10 SummationDelivered</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0113	<i>CurrentTier10 SummationReceived</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0114	<i>CurrentTier11 SummationDelivered</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O

0x0115	<i>CurrentTier11 SummationReceived</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0116	<i>CurrentTier12 SummationDelivered</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0117	<i>CurrentTier12 SummationReceived</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0118	<i>CurrentTier13 SummationDelivered</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0119	<i>CurrentTier13 SummationReceived</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x011A	<i>CurrentTier14 SummationDelivered</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x011B	<i>CurrentTier14 SummationReceived</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x011C	<i>CurrentTier15 SummationDelivered</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x011D	<i>CurrentTier15 SummationReceived</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x011E	<i>CurrentTier16 SummationDelivered</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x011F	<i>CurrentTier16 SummationReceived</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0120	<i>CurrentTier17 SummationDelivered</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0121	<i>CurrentTier17 SummationReceived</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
...
0x015E	<i>CurrentTier48 SummationDelivered</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x015F	<i>CurrentTier48 SummationReceived</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O

0x0160 to 0x01FB	Reserved					
0x01FC	<i>CPP1 Summation Delivered</i>	Unsigned 48-bit integer	0x00000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x01FD	Reserved					
0x01FE	<i>CPP2 Summation Delivered</i>	Unsigned 48-bit integer	0x00000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x01FF	Reserved					

5582

5583 D.3.2.2.1 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 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.3.2.2.3 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.3.2.2.4 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

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	<i>Status</i>	8-bit BitMap	0x00 to 0xFF	Read Only	0x00	M
0x0201	<i>Remaining BatteryLife</i>	Unsigned 8-bit Integer	0x00 to 0xFF	Read Only	-	O
0x0202	<i>HoursIn Operation</i>	Unsigned 24bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O ¹⁰
0x0203	<i>HoursIn Fault</i>	Unsigned 24bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0204	<i>Extended Status</i>	64-bit BitMap	0x0000000000000000 to 0xFFFFFFFFFFFFFF	Read Only	-	O
0x0205	<i>Remaining BatteryLife in Days</i>	Unsigned 16bit Integer	0x0000 to 0xFFFF	Read Only	-	O
0x0206	<i>CurrentMeter ID</i>	Octet String		Read Only	-	O
0x0207	<i>Ambient Consumption Indicator</i>	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

5622 available to upstream systems together with consumption data collected from the battery
5623 operated meter.

5624 **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

5625
5626 The definitions of the Electricity *Status* bits are:

5627 **Service Disconnect Open:** Set to true when the service has been disconnected to this premises.

5628 **Leak Detect:** Set to true when a leak has been detected.

5629 **Power Quality:** Set to true if a power quality event has been detected such as a low voltage,
5630 high voltage.

5631 **Power Failure:** Set to true during a power outage.

5632 **Tamper Detect:** Set to true if a tamper event has been detected.

5633 **Low Battery:** Set to true when the battery needs maintenance.

5634 **Check Meter:** Set to true when a non fatal problem has been detected on the meter such as a
5635 measurement error, memory error, self check error.

5636 **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

5637
5638 The definitions of the Gas *Status* bits are:

5639 **Reverse Flow:** Set to true if flow detected in the opposite direction to normal (from consumer
5640 to supplier).

5641 **Service Disconnect:** Set to true when the service has been disconnected to this premises. Ex.
5642 The valve is in the closed position preventing delivery of gas.

5643 **Leak Detect:** Set to true when a leak has been detected.

5644 **Low Pressure:** Set to true when the pressure at the meter is below the meter's low pressure

5645 threshold value.

5646 **Tamper Detect:** Set to true if a tamper event has been detected.

5647 **Low Battery:** Set to true when the battery needs maintenance.

5648 **Check Meter:** Set to true when a non fatal problem has been detected on the meter such as a
5649 measurement error, memory error, or self check error.

5650 **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

5651

5652 The definitions of the Water *Status* bits are:

5653 **Reverse Flow:** Set to true if flow detected in the opposite direction to normal (from consumer
5654 to supplier).

5655 **Service Disconnect:** Set to true when the service has been disconnected to this premises. Ex.
5656 The valve is in the closed position preventing delivery of water.

5657 **Leak Detect:** Set to true when a leak has been detected.

5658 **Low Pressure:** Set to true when the pressure at the meter is below the meter's low pressure
5659 threshold value.

5660 **Pipe Empty:** Set to true when the service pipe at the meter is empty and there is no flow in
5661 either direction.

5662 **Tamper Detect:** Set to true if a tamper event has been detected.

5663 **Low Battery:** Set to true when the battery needs maintenance.

5664 **Check Meter:** Set to true when a non fatal problem has been detected on the meter such as a
5665 measurement error, memory error, or self check error.

5666 **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

5667

- 5668 The definitions of the Heat and Cooling *Status* bits are:
- 5669 **Flow Sensor:** Set to true *when an error is detected on a flow sensor at this premises.*
- 5670 **Service Disconnect:** Set to true when the service has been disconnected to this premises. Ex.
5671 The valve is in the closed position preventing delivery of heat or cooling.
- 5672 **Leak Detect:** Set to true when a leak has been detected.
- 5673 **Burst Detect:** Set to true *when a burst is detected on pipes at this premises.*
- 5674 **Temperature Sensor:** Set to true *when an error is detected on a temperature sensor* at this
5675 premises.
- 5676 **Tamper Detect:** Set to true if a tamper event has been detected.
- 5677 **Low Battery:** Set to true when the battery needs maintenance.
- 5678 **Check Meter:** Set to true when a non fatal problem has been detected on the meter such as a
5679 measurement error, memory error, or self check error.
- 5680 *Note: It is not necessary to set aside Bit 7 as an “Extension Bit” for future expansion. If*
5681 *extra status bits are required an Extended Meter Status attribute may be added to support*
5682 *additional status values.*
- 5683 D.3.2.2.3.2 RemainingBatteryLife Attribute
- 5684 *RemainingBatteryLife* *represents the estimated remaining life of the battery in % of capacity.* A
5685 setting of *0xFF indicates this feature is disabled.* The range 0 - 100 where 100 = 100%, *0xFF =*
5686 *Unknown.*
- 5687 D.3.2.2.3.3 HoursInOperation Attribute
- 5688 *HoursInOperation* is a *counter that increments once every hour during operation.* This may be
5689 used as a check for tampering.
- 5690 *Note: For meters that are not electricity meters turning off the meter does not necessarily*
5691 *prevent delivery of energy — but the meter might not be able to measure it.*
- 5692 D.3.2.2.3.4 HoursInFault Attribute
- 5693 *HoursInFault* is a *counter that increments once every hour when the device is in operation with*
5694 *a fault detected.* This may be used as a check for tampering.
- 5695 *Note: For meters that are not electricity meters turning off the meter does not necessarily*
5696 *prevent delivery of energy - but the meter might not be able to measure it.*

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-volatile) memory.

5714 **RAM Error:** Set to true when the device detects an instance of a Random Access Memory (RAM) error within the device memory.

5716 **NV Memory Error:** Set to true when the device detects an instance of a Non Volatile (NV)
 5717 memory error within the device memory - this is a fatal meter error that will require the meter
 5718 replacement.

5719 **Measurement System Error:** Set to true when the device detects an error within its
 5720 measurement system.

5721 **Watchdog Error:** Set to true when the device has detected an instance of a watchdog reset
 5722 event (following a catastrophic fault within the device).

5723 **Supply Disconnect Failure:** Set to true when the device has detected that the valve has not
 5724 closed as expected (for gas) or the contactor has not opened as expected (for electricity).

5725 **Supply Connect Failure:** Set to true when the device has detected that the valve has not opened
 5726 as expected (for gas) or the contactor has not closed as expected (for electricity).

5727 **Measurement SW Changed/Tampered:** Set to true when the device detects that its
 5728 measurement software has changed.

5729 **Clock Invalid:** Set to true when the device detects that its internal clock is invalid.

5730 **Temperature Exceeded:** Set to true when the metering device's temperature exceeds a
 5731 predefined limit. There are various reasons for temperature rise in metering devices.

5732 **Moisture Detected:** Set to true when a sensor has detected the presence of moisture e.g. moisture
 5733 in a gas line which can cause a drop in gas pressure, or moisture detected in the sealed
 5734 component area within a water meter.

5735

5736 **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

5737

5738 The definitions of the Electricity-Meter-Specific *Extended Status* bits are:

5739 **Terminal Cover Removed:** Set to true when the device detects that its terminal cover has been
5740 removed.

5741 Incorrect Polarity: Set to true when the electricity meter detects incorrect polarity on the
5742 electricity supply.

5743 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.

5745 **Limit Threshold Exceeded:** Set to true when the electricity meter detects that the load has
5746 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.

5751

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

5753
5754 The definitions of the Gas-Meter-Specific *Extended Status* bits are:

5755 Battery Cover Removed: Set to true when the gas meter detects that its battery cover has been
5756 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).

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 – 0xFFFF, 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*)

5790 **Demand = Demand received * Multiplier / Divisor**

5791 (formatted using *DemandFormatting*)

5792 **Temperature = Temperature received * Multiplier / Divisor**

5793 If the Multiplier and Divisor attribute values are zero, just the formatting hints defined in
 5794 *SummationFormatting*, *ConsumptionFormatting*, *DemandFormatting* and
 5795 *TemperatureFormatting* attributes are used.

5796 The summation received, consumption received, demand received, and temperature received
 5797 variables used above can be replaced by any of the attributes listed in sub-clauses
 5798 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.

5799 The following table shows examples that demonstrate the relation between these attributes.

5800

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

5801

5802 The Consumption Formatting Attribute Set is defined in Table D-25.

5803

5804

Table D-25– Formatting Attribute Set

Identifier	Name	Type	Range	Access	Default	Man./Opt.
0x0300	<i>UnitofMeasure</i>	8-bit Enumeration	0x00 to 0xFF	Read Only	0x00	M
0x0301	<i>Multiplier</i>	Unsigned 24-bit Integer	0x000000 0 to 0xFFFFFFFF	Read Only	-	O

0x0302	<i>Divisor</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFFF	Read Only	-	O
0x0303	<i>SummationFormatting</i>	8-bit BitMap	0x00 to 0xFF	Read Only	-	M
0x0304	<i>DemandFormatting</i>	8-bit BitMap	0x00 to 0xFF	Read Only	-	O
0x0305	<i>HistoricalConsumptionFormatting</i>	8-bit BitMap	0x00 to 0xFF	Read Only	-	O
0x0306	<i>MeteringDeviceType</i>	8-bit BitMap	0x00 to 0xFF	Read Only	-	M
0x0307	<i>SiteID</i>	Octet String	1 to 33 Octets	Read only	-	O
0x0308	<i>MeterSerialNumber</i>	Octet String	1 to 25 Octets	Read only	-	O
0x0309	<i>EnergyCarrierUnitOfMeasure</i>	8-bit Enumeration	0x00 to 0xFF	Read Only	-	O ¹¹
0x030A	<i>EnergyCarrierSummationFormatting</i>	8-bit BitMap	0x00 to 0xFF	Read Only	-	O ¹²
0x030B	<i>EnergyCarrierDemandFormatting</i>	8-bit BitMap	0x00 to 0xFF	Read Only	-	O
0x030C	<i>TemperatureUnitOfMeasure</i>	8-bit Enumeration	0x00 to 0xFF	Read Only	-	O ¹³
0x030D	<i>TemperatureFormatting</i>	8-bit BitMap	0x00 to 0xFF	Read Only	-	O ¹⁴
0x030E	<i>ModuleSerialNumber</i>	Octet String	1 to 25 Octets	Read only	-	O
0x030F	<i>OperatingTariffLabelDelivered</i>	Octet String	1 to 25 Octets	Read only	-	O
0x0310	<i>OperatingTariffLabelReceived</i>	Octet String	1 to 25 Octets	Read only	-	O
0x0311	<i>CustomerIDNumber</i>	Octet String	1 to 25 Octets	Read only	-	O
0x0312	<i>AlternativeUnitofMeasure</i>	8-bit Enumeration	0x00 to 0xFF	Read Only	0x00	O

¹¹ CCB 1999

¹² CCB 1999

¹³ CCB 1999

¹⁴ CCB 1999

0x0313	<i>Alternative DemandFormatting</i>	8-bit BitMap	0x00 to 0xFF	Read Only	-	O
0x0314	<i>Alternative ConsumptionFormatting</i>	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.

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.

5815

5816 **Note:** When using BCD for meter reads, the values A to F are special values or indicators
 5817 denoting “Opens”, “Shorts”, and etc. conditions when reading meter register hardware. Any
 5818 SE device displaying the BCD based values to end users should use a non-decimal value to
 5819 replace the A to F. In other words, a device could use an “*” in place of the special values or
 5820 indicators.

5821 D.3.2.2.4.2 Multiplier Attribute

5822 **Multiplier** provides a value to be multiplied against a raw or uncompensated sensor count of
 5823 Energy, Gas, or Water being measured by the metering device. If present, this attribute must be
 5824 applied against all summation, consumption and demand values to derive the delivered and
 5825 received values expressed in the unit of measure specified. **This attribute must be used in**
 5826 **conjunction with the Divisor attribute.**

5827 D.3.2.2.4.3 Divisor Attribute

5828 **Divisor** provides a value to divide the results of applying the Multiplier Attribute against a raw
 5829 or uncompensated sensor count of Energy, Gas, or Water being measured by the metering
 5830 device. If present, this attribute must be applied against all summation, consumption and
 5831 demand values to derive the delivered and received values expressed in the unit of measure
 5832 specified. **This attribute must be used in conjunction with the Multiplier attribute.**

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:

5861 **Bits 0 to 2:** Number of Digits to the right of the Decimal Point.

5862 **Bits 3 to 6:** Number of Digits to the left of the Decimal Point.

5863 **Bit 7:** If set, suppress leading zeros.

5864 This attribute shall be used against the following attributes:

5865 • *CurrentDayConsumptionDelivered*

5866 • *CurrentDayConsumptionReceived*

5867 • *PreviousDayConsumptionDelivered*

5868 • *PreviousDayConsumptionReceived*

5869 • *CurrentPartialProfileIntervalValue*

5870 • *Intervals*

5871 • *DailyConsumptionTarget*

5872 • *CurrentDayConsumptionDelivered*

5873 • *CurrentDayConsumptionReceived*

5874 • *PreviousDayNConsumptionDelivered*

5875 • *PreviousDayNConsumptionReceived*

5876 • *CurrentWeekConsumptionDelivered*

5877 • *CurrentWeekConsumptionReceived*

5878 • *PreviousWeekNConsumptionDelivered*

5879 • *PreviousWeekNConsumptionReceived*

5880 • *CurrentMonthConsumptionDelivered*

5881 • *CurrentMonthConsumptionReceived*

5882 • *PreviousMonthNConsumptionDelivered*

5883 • *PreviousMonthNConsumptionReceived*

5884 D.3.2.2.4.7 MeteringDeviceType Attribute

5885 *MeteringDeviceType* provides a label for identifying the type of metering device present. The
5886 attribute are values representing Energy, Gas, Water, Thermal, Heat, Cooling, and mirrored
5887 metering devices. The defined values are represented in Table D-27. (Note that these values
5888 represent an Enumeration, and not an 8-bit BitMap as indicated in the attribute description.)

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

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

5896 **Note:** Heat and cooling meters are used for measurement and billing of heat (and cooling)
 5897 delivered through liquid (water) based central heating systems. The consumers are typically
 5898 billed by the kWh, calculated from the flow and the temperatures in and out.

5899 D.3.2.2.4.8 SiteID Attribute

5900 The *SiteID* is a ZCL Octet String field capable of storing a 32 character string (the first Octet
 5901 indicates length) encoded in UTF-8 format. The *SiteID* is a text string, known in the UK as the
 5902 MPAN number for electricity, MPRN for gas and 'Stand Point' in South Africa. These numbers
 5903 specify the meter point location in a standardized way. The field is defined to accommodate
 5904 the number of characters typically found in the UK and Europe (16 digits). Generally speaking
 5905 the field is numeric but is defined for the possibility of an alpha-numeric format by
 5906 specifying an octet string.

5907 D.3.2.2.4.9 MeterSerialNumber Attribute

5908 The *MeterSerialNumber* is a ZCL Octet String field capable of storing a 24 character string
 5909 (the first Octet indicates length) encoded in UTF-8 format. It is used to provide a unique
 5910 identification of the metering device.

5911 D.3.2.2.4.10 EnergyCarrierUnitOfMeasure Attribute

5912 The *EnergyCarrierUnitOfMeasure* specifies the unit of measure that the *EnergyCarrier* is
 5913 measured in. This unit of measure is typically a unit of volume or flow and cannot be an
 5914 amount of energy. The enumeration of this attribute is otherwise identical to the *UnitofMeasure*
 5915 attribute (Table D-26).

5916 D.3.2.2.4.11 EnergyCarrierSummationFormatting Attribute

5917 *EnergyCarrierSummationFormatting* provides a method to properly decipher the number of
 5918 digits and the decimal location of the values found in the Summation- related attributes.

5919 This attribute is to be decoded as follows:

5920 Bits 0 to 2: Number of Digits to the right of the Decimal Point.

5921 Bits 3 to 6: Number of Digits to the left of the Decimal Point.

5922 Bit 7: If set, suppress leading zeros.

5923 This attribute shall be used in relation with the following attributes:

- *CurrentInletEnergyCarrierSummation*
 - *CurrentOutletEnergyCarrierSummatio*

5926 D.3.2.4.12 EnergyCarrierDemandFormatting Attribute

5927 *EnergyCarrierDemandFormatting* provides a method to properly decipher the number of
5928 digits and the decimal location of the values found in the Demand- related attributes.

5929 This attribute is to be decoded as follows:

5930 Bits 0 to 2: Number of Digits to the right of the Decimal Point.

5931 Bits 3 to 6: Number of Digits to the left of the Decimal Point.

5932 Bit 7: If set, suppress leading zeros.

5933 This attribute shall be used in relation with the following attributes:

- *CurrentInletEnergyCarrierDemand*
 - *CurrentOutletEnergyCarrierDemand*
 - *CurrentDayMaxEnergyCarrierDemand*
 - *PreviousDayMaxEnergyCarrierDemand*
 - *CurrentMonthMaxEnergyCarrierDemand*
 - *CurrentMonthMinEnergyCarrierDemand*
 - *CurrentYearMinEnergyCarrierDemand*
 - *CurrentYearMaxEnergyCarrierDemand*

5942 D.3.2.2.4.13 TemperatureUnitOfMeasure Attribute

5943 The *TemperatureUnitOfMeasure* specifies the unit of measure that temperatures are measured
5944 in. The enumeration of this attribute is as follows.

Table D-28– TemperatureUnitOfMeasure Enumeration

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.

5969 D.3.2.2.4.17 OperatingTariffLabelReceived Attribute

5970 The *OperatingTariffLabelReceived* attribute is the meter's version of the *ReceivedTariffLabel*
5971 attribute that is found within the Tariff Information attribute set of the Price Cluster. It is used to
5972 identify the current generation tariff operating on the meter. See section D.4.2.2.15.1. The
5973 attribute is a ZCL Octet String field capable of storing a 24 character string (the first Octet
5974 indicates length) encoded in UTF-8 format.

5975 D.3.2.2.4.18 CustomerIDNumber Attribute

5976 The *CustomerIDNumber* attribute provides a customer identification which may be used to
5977 confirm the customer at the premises. The attribute is a ZCL Octet String field capable of storing
5978 a 24 character string (not including the first Octet which indicates length) encoded in UTF-8
5979 format.

5980 D.3.2.2.4.19 AlternativeUnitofMeasure Attribute

5981 Unless stated otherwise, the *AlternativeUnitofMeasure* attribute provides a base for the attributes
5982 in the Alternative Historical Consumption attribute set defined in Table D-46.

5983 The *AlternativeUnitofMeasure* attribute shall be supported if any of the attributes within the
5984 Alternative Historical Consumption attribute set are to be used.

5985 The *AlternativeUnitofMeasure* attribute shall be set to a value that is different to the
5986 *UnitOfMeasure* attribute.

5987 The *AlternativeUnitofMeasure* attribute is an 8-bit enumerated field. The possible values for this
5988 attribute are listed in Table D-26.

5989 D.3.2.2.4.20 AlternativeDemandFormatting Attribute

5990 *AlternativeDemandFormatting* provides a method to properly decipher the number of digits and
5991 the decimal location of the values found in the Alternative Demand-related attributes. This
5992 attribute is to be decoded as follows:

5993 **Bits 0 to 2:** Number of Digits to the right of the Decimal Point.

5994 **Bits 3 to 6:** Number of Digits to the left of the Decimal Point.

5995 **Bit 7:** If set, suppress leading zeros.

5996 This attribute shall be used against the following attribute:

- 5997 • *AlternativeInstantaneousDemand*

5998

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

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	<i>InstantaneousDemand</i>	Signed 24-bit Integer	-8,388,607 to 8,388,607	Read Only	0x00	O
0x0401	<i>CurrentDayConsumption Delivered</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0402	<i>CurrentDayConsumption Received</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0403	<i>PreviousDayConsumption Delivered</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0404	<i>PreviousDayConsumption Received</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0405	<i>CurrentPartialProfile IntervalStartTimeDelivered</i>	UTCTime		Read Only	-	O
0x0406	<i>CurrentPartialProfile IntervalStartTimeReceived</i>	UTCTime	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0407	<i>CurrentPartialProfile IntervalValueDelivered</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0408	<i>CurrentPartialProfile IntervalValueReceived</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0409	<i>CurrentDayMaxPressure</i>	Unsigned 48-bit Integer	0x000000 000000 to 0xFFFFFFF FFFFFFF	Read Only	-	O
0x040A	<i>CurrentDayMinPressure</i>	Unsigned 48-bit Integer	0x000000 000000 to 0xFFFFFFF FFFFFFF	Read Only	-	O
0x040B	<i>PreviousDayMaxPressure</i>	Unsigned 48-bit Integer	0x000000 000000 to 0xFFFFFFF FFFFFFF	Read Only	-	O

0x040C	<i>PreviousDayMinPressure</i>	Unsigned 48-bit Integer	0x000000 000000 to 0xFFFFFFF FFFFFFF	Read Only	-	O
0x040D	<i>CurrentDayMaxDemand</i>	Signed 24-bit Integer	-8,388,607 to 8,388,607	Read Only	-	O
0x040E	<i>PreviousDayMaxDemand</i>	Signed 24-bit Integer	-8,388,607 to 8,388,607	Read Only	-	O
0x040F	<i>CurrentMonthMax Demand</i>	Signed 24-bit Integer	-8,388,607 to 8,388,607	Read Only	-	O
0x0410	<i>CurrentYearMaxDemand</i>	Signed 24-bit Integer	-8,388,607 to 8,388,607	Read Only	-	O
0x0411	<i>CurrentDayMaxEnergy CarrierDemand</i>	Signed 24-bit integer	-8,388,607 to 8,388,607	Read Only	-	O
0x0412	<i>PreviousDayMaxEnergy CarrierDemand</i>	Signed 24-bit integer	-8,388,607 to 8,388,607	Read Only	-	O
0x0413	<i>CurrentMonthMax EnergyCarrierDemand</i>	Signed 24-bit Integer	-8,388,607 to 8,388,607	Read Only	-	O
0x0414	<i>CurrentMonthMinEnergy CarrierDemand</i>	Signed 24-bit Integer	-8,388,607 to 8,388,607	Read Only	-	O
0x0415	<i>CurrentYearMaxEnergy CarrierDemand</i>	Signed 24-bit Integer	-8,388,607 to 8,388,607	Read Only	-	O
0x0416	<i>CurrentYearMinEnergy CarrierDemand</i>	Signed 24-bit Integer	-8,388,607 to 8,388,607	Read Only	-	O
0x0417 to 0x041F	Reserved					
0x0420	<i>PreviousDay2Consumption Delivered</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0421	<i>PreviousDay2Consumption Received</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0422	<i>PreviousDay3Consumption Delivered</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0423	<i>PreviousDay3Consumption Received</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O

0x0424	<i>PreviousDay4Consumption Delivered</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0425	<i>PreviousDay4Consumption Received</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0426	<i>PreviousDay5Consumption Delivered</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0427	<i>PreviousDay5Consumption Received</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0428	<i>PreviousDay6Consumption Delivered</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0429	<i>PreviousDay6Consumption Received</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x042A	<i>PreviousDay7Consumption Delivered</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x042B	<i>PreviousDay7Consumption Received</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x042C	<i>PreviousDay8Consumption Delivered</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x042D	<i>PreviousDay8Consumption Received</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x042E to 0x042F	Reserved					
0x0430	<i>CurrentWeekConsumption Delivered</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0431	<i>CurrentWeekConsumption Received</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0432	<i>PreviousWeekConsumption Delivered</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0433	<i>PreviousWeekConsumption Received</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O

0x0434	<i>PreviousWeek2Consumption Delivered</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0435	<i>PreviousWeek2Consumption Received</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0436	<i>PreviousWeek3Consumption Delivered</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0437	<i>PreviousWeek3Consumption Received</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0438	<i>PreviousWeek4Consumption Delivered</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0439	<i>PreviousWeek4Consumption Received</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x043A	<i>PreviousWeek5Consumption Delivered</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x043B	<i>PreviousWeek5Consumption Received</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x043C to 0x043F	Reserved					
0x0440	<i>CurrentMonthConsumption Delivered</i>	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0441	<i>CurrentMonthConsumption Received</i>	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0442	<i>PreviousMonthConsumption Delivered</i>	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0443	<i>PreviousMonthConsumption Received</i>	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0444	<i>PreviousMonth2Consumption Delivered</i>	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0445	<i>PreviousMonth2Consumption Received</i>	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0446	<i>PreviousMonth3Consumption Delivered</i>	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O

0x0447	<i>PreviousMonth3Consumption Received</i>	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0448	<i>PreviousMonth4Consumption Delivered</i>	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0449	<i>PreviousMonth4Consumption Received</i>	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x044A	<i>PreviousMonth5Consumption Delivered</i>	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x044B	<i>PreviousMonth5Consumption Received</i>	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x044C	<i>PreviousMonth6Consumption Delivered</i>	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x044D	<i>PreviousMonth6Consumption Received</i>	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x044E	<i>PreviousMonth7Consumption Delivered</i>	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x044F	<i>PreviousMonth7Consumption Received</i>	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0450	<i>PreviousMonth8Consumption Delivered</i>	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0451	<i>PreviousMonth8Consumption Received</i>	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0452	<i>PreviousMonth9Consumption Delivered</i>	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0453	<i>PreviousMonth9Consumption Received</i>	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0454	<i>PreviousMonth10Consumption Delivered</i>	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0455	<i>PreviousMonth10Consumption Received</i>	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0456	<i>PreviousMonth11Consumption Delivered</i>	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O

0x0457	<i>PreviousMonth11Consumption Received</i>	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0458	<i>PreviousMonth12Consumption Delivered</i>	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0459	<i>PreviousMonth12Consumption Received</i>	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x045A	<i>PreviousMonth13Consumption Delivered</i>	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x045B	<i>PreviousMonth13Consumption Received</i>	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x045C	<i>Historical Freeze Time</i>	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.

6047 D.3.2.2.5.4 PreviousDayConsumptionDelivered Attribute

6048 *PreviousDayConsumptionDelivered* represents the summed value of Energy, Gas, or Water
6049 delivered to the premises within the previous 24 hour period starting at the *Historical Freeze*
6050 *Time* (HFT). If optionally provided, *PreviousDayConsumptionDelivered* is updated every HFT.
6051 If the optional HFT attribute is not available, default to midnight local time.

6052 D.3.2.2.5.5 PreviousDayConsumptionReceived Attribute

6053 *PreviousDayConsumptionReceived* represents the summed value of Energy, Gas, or Water
6054 received from the premises within the previous 24 hour period starting at the *Historical Freeze*
6055 *Time* (HFT). If optionally provided, *PreviousDayConsumptionReceived* is updated every HFT. If
6056 the optional *HFT* attribute is not available, default to midnight local time.

6057 D.3.2.2.5.6 CurrentPartialProfileIntervalStartTimeDelivered Attribute

6058 *CurrentPartialProfileIntervalStartTimeDelivered* represents the start time of the current Load
6059 Profile interval being accumulated for commodity delivered.

6060 D.3.2.2.5.7 CurrentPartialProfileIntervalStartTimeReceived Attribute

6061 *CurrentPartialProfileIntervalStartTimeReceived* represents the start time of the current Load
6062 Profile interval being accumulated for commodity received.

6063 D.3.2.2.5.8 CurrentPartialProfileIntervalValueDelivered Attribute

6064 *CurrentPartialProfileIntervalValueDelivered* represents the value of the current Load Profile
6065 interval being accumulated for commodity delivered.

6066 D.3.2.2.5.9 CurrentPartialProfileIntervalValueReceived Attribute

6067 *CurrentPartialProfileIntervalValueReceived* represents the value of the current Load Profile
6068 interval being accumulated for commodity received.

6069 D.3.2.2.5.10 CurrentDayMaxPressure Attribute

6070 *CurrentDayMaxPressure* is the maximum pressure reported during a day from the water or gas
6071 meter.

6072 D.3.2.2.5.11 PreviousDayMaxPressure Attribute

6073 *PreviousDayMaxPressure* represents the maximum pressure reported during previous day from
6074 the water or gas meter.

6075 D.3.2.2.5.12 CurrentDayMinPressure Attribute

6076 *CurrentDayMinPressure* is the minimum pressure reported during a day from the water or gas
6077 meter.

6078 D.3.2.2.5.13 PreviousDayMinPressure Attribute

6079 *PreviousDayMinPressure* represents the minimum pressure reported during previous day from
6080 the water or gas meter.

6081 D.3.2.2.5.14 CurrentDayMaxDemand Attribute

6082 *CurrentDayMaxDemand* represents the maximum demand or rate of delivered value of
6083 Energy, Gas, or Water being utilized at the premises.

6084 D.3.2.2.5.15 PreviousDayMaxDemand Attribute

6085 *PreviousDayMaxDemand* represents the maximum demand or rate of delivered value of
6086 Energy, Gas, or Water being utilized at the premises.

6087 *Note: At the end of a day the metering device will transfer the CurrentDayMaxPressure into*
6088 *PreviousDayMaxPressure, CurrentDayMinPressure into PreviousDayMinPressure and*
6089 *CurrentDayMaxDemand into PreviousDayMaxDemand.*

6090 D.3.2.2.5.16 CurrentMonthMaxDemand Attribute

6091 *CurrentMonthMaxDemand* is the maximum demand reported during a month from the meter.

6092 For electricity, heat and cooling meters this is the maximum power reported in a month.

6093 D.3.2.2.5.17 CurrentYearMaxDemand Attribute

6094 *CurrentYearMaxDemand* is the maximum demand reported during a year from the meter.

6095 For electricity, heat and cooling meters this is the maximum power reported in a year.

6096 D.3.2.2.5.18 CurrentDayMaxEnergyCarrierDemand Attribute

6097 *CurrentDayMaxEnergyCarrierDemand* is the maximum energy carrier demand reported during
6098 a day from the meter.

6099 *Note: At the end of a day the meter will transfer the CurrentDayMaxEnergyCarrierDemand into*
6100 *PreviousDayMaxEnergyCarrierDemand.*

6101 For heat and cooling meters this is the maximum flow rate on the inlet reported in a day.

6102 D.3.2.2.5.19 PreviousDayMaxEnergyCarrierDemand Attribute

6103 *PreviousDayMaxEnergyCarrierDemand* is the maximum energy carrier demand reported during
6104 the previous day from the meter.

6105 D.3.2.2.5.20 CurrentMonthMaxEnergyCarrierDemand Attribute

6106 *CurrentMonthMaxEnergyCarrierDemand* is the maximum energy carrier demand reported during
6107 a month from the meter.

6108 For heat and cooling meters this is the maximum flow rate on the inlet reported in a month.

6109 D.3.2.2.5.21 CurrentMonthMinEnergyCarrierDemand Attribute

6110 *CurrentMonthMinEnergyCarrierDemand* is the minimum energy carrier demand reported during
6111 a month from the meter.

6112 For heat and cooling meters this is the minimum flow rate on the inlet reported in a month.

6113 *Note:* This attribute may be used to detect leaks if there has been no flow rate of zero in the last
6114 month.

6115 D.3.2.2.5.22 CurrentYearMaxEnergyCarrierDemand Attribute

6116 *CurrentYearMaxEnergyCarrierDemand* is the maximum energy carrier demand reported during
6117 a year from the meter.

6118 For heat and cooling meters this is the maximum flow rate on the inlet reported in a year.

6119 D.3.2.2.5.23 CurrentYearMinEnergyCarrierDemand Attribute

6120 *CurrentYearMinEnergyCarrierDemand* is the minimum energy carrier demand reported during
6121 a year from the heat meter.

6122 For heat and cooling meters this is the minimum flow rate on the inlet reported in a year.

6123 *Note:* This attribute may be used to detect leaks if there has been no flow rate of zero in the last
6124 year

6125 D.3.2.2.5.24 PreviousDayNConsumptionDelivered Attribute

6126 *PreviousDayNConsumptionDelivered* represents the summed value of Energy, Gas, or Water
6127 delivered to the premises within the previous 24 hour period starting at the *Historical Freeze*
6128 *Time* (HFT). If the optional HFT attribute is not available, default to midnight local time.

6129 D.3.2.2.5.25 PreviousDayNConsumptionReceived Attribute

6130 *PreviousDayNConsumptionReceived* represents the summed value of Energy, Gas, or Water
6131 received from the premises within the previous 24 hour period starting at the *Historical Freeze*
6132 *Time* (HFT). If the optional HFT attribute is not available, default to midnight local time.

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.

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	<i>MaxNumberOfPeriods Delivered</i>	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,

6195 or an optionally agreed upon larger Fragmentation ASDU size supported by both devices.
6196 Please refer to sub-clause 5.3.8 for further details on Fragmentation settings.

6197

6198 D.3.2.2.7 **Supply Limit Attributes**

6199 This set of attributes is used to implement a “Supply Capacity Limit” program where the
6200 demand at the premises is limited to a preset consumption level over a preset period of time.
6201 Should this preset limit be exceeded the meter could interrupt supply to the premises or to
6202 devices within the premises. The supply limit information in this attribute set can be used by
6203 In-Home displays, PCTs, or other devices to display a warning when the supply limit is
6204 being approached. The Supply Limit Attribute Set is defined in Table D-31.

6205

Table D-31– Supply Limit Attribute Set

Identifier	Name	Type	Range	Access	Default	Man / Opt
0x0600	<i>CurrentDemand Delivered</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read only		O
0x0601	<i>DemandLimit</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read only		O
0x0602	<i>DemandIntegration Period</i>	Unsigned 8-bit Integer	0x01 to 0xFF	Read only	-	O
0x0603	<i>NumberOfDemand Subintervals</i>	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					

6206

6207 D.3.2.2.7.1 CurrentDemandDelivered Attribute

6208 *CurrentDemandDelivered* represents the current Demand of Energy, Gas, or Water delivered at
6209 the premises. *CurrentDemandDelivered* may be continuously updated as new measurements are
6210 acquired, but at a minimum *CurrentDemandDelivered* must be updated at the end of each
6211 integration sub-period, which can be obtained by dividing the *DemandIntegrationPeriod* by
6212 the *NumberOfDemandSubintervals*.

6213 This attribute shall be adjusted using the *Multiplier* and *Divisor* attributes found in the Formatting
6214 Attribute Set and can be formatted using the *DemandFormatting* attribute. The final result
6215 represents an engineering value in the unit defined by the *UnitofMeasure* attribute.

6216 D.3.2.2.7.2 DemandLimit Attribute

6217 *DemandLimit* reflects the current supply demand limit set in the meter. This value can be
6218 compared to the *CurrentDemandDelivered* attribute to understand if limits are being approached
6219 or exceeded.

6220 Adjustment and formatting of this attribute follow the same rules as the
6221 *CurrentDemandDelivered*.

6222 A value of “0xFFFF” indicates “demand limiting” is switched off.

6223 D.3.2.2.7.3 DemandIntegrationPeriod Attribute

6224 *DemandIntegrationPeriod* is the number of minutes over which the *CurrentDemandDelivered*
6225 attribute is calculated. Valid range is 0x01 to 0xFF. 0x00 is a reserved value.

6226 D.3.2.2.7.4 NumberOfDemandSubintervals Attribute

6227 *NumberOfDemandSubintervals* represents the number of subintervals used within the
6228 *DemandIntegrationPeriod*. The subinterval duration (in minutes) is obtained by dividing the
6229 *DemandIntegrationPeriod* by the *NumberOfDemandSubintervals*. The *CurrentDemandDelivered*
6230 attribute is updated at the end of each subinterval. Valid range is 0x01 to 0xFF. 0x00 is a
6231 reserved value.

6232 As a Rolling Demand example, *DemandIntegrationPeriod* could be set at 30 (for 30 minute
6233 period) and *NumberOfDemandSubintervals* could be set for 6. This would provide 5 minute
6234 (30/6 = 5) subinterval periods.

6235 As a Block Demand example, *DemandIntegrationPeriod* could be set at 30 (for 30 minute
6236 period) and *NumberOfDemandSubintervals* could be set for 1. This would provide a single 30
6237 minute subinterval period.

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.

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	<i>CurrentNoTier Block1Summation Delivered</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0701	<i>CurrentNoTier Block2Summation Delivered</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0702	<i>CurrentNoTier Block3Summation Delivered</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x070N	... <i>CurrentNoTier BlockN+1 SummationDelivered</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x070F	<i>CurrentNoTier Block16Summation Delivered</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0710	<i>CurrentTier1 Block1Summation Delivered</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0711	<i>CurrentTier1 Block2Summation Delivered</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0712	<i>CurrentTier1 Block3Summation Delivered</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x071N	<i>CurrentTier1 BlockN+1 SummationDelivered</i> ...	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x071F	<i>CurrentTier1 Block16Summation Delivered</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0720	<i>CurrentTier2 Block1Summation Delivered</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O

0x072N	<i>CurrentTier2 BlockN+1 SummationDelivered</i> ...	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x072F	<i>CurrentTier2 Block16Summation Delivered</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0730	<i>CurrentTier3 Block1Summation Delivered</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x073N	<i>CurrentTier3 BlockN+1 SummationDelivered</i> ...	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x073F	<i>CurrentTier3 Block16Summation Delivered</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0740	<i>CurrentTier4 Block1Summation Delivered</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x074N	<i>CurrentTier4 BlockN+1 SummationDelivered</i> ...	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x074F	<i>CurrentTier4 Block16 SummationDelivered</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0750	<i>CurrentTier5 Block1Summation Delivered</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x075N	<i>CurrentTier5 BlockN+1 SummationDelivered</i> ...	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x075F	<i>CurrentTier5 Block16Summation Delivered</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0760	<i>CurrentTier6 Block1Summation Delivered</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x076N	<i>CurrentTier6 BlockN+1 SummationDelivered</i> ...	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O

0x076F	<i>CurrentTier6 Block16Summation Delivered</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0770	<i>CurrentTier7 Block1Summation Delivered</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x077N	<i>CurrentTier7 BlockN+1 SummationDelivered</i> ...	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x077F	<i>CurrentTier7 Block16Summation Delivered</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0780	<i>CurrentTier8 Block1Summation Delivered</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x078N	<i>CurrentTier8 BlockN+1 SummationDelivered</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x078F	<i>CurrentTier8 Block16Summation Delivered</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0790	<i>CurrentTier9 Block1Summation Delivered</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x079N	<i>CurrentTier9 BlockN+1 SummationDelivered</i> ...	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x079F	<i>CurrentTier9 Block16Summation Delivered</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x07A0	<i>CurrentTier10 Block1Summation Delivered</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x07AN	<i>CurrentTier10 BlockN+1 SummationDelivered</i> ...	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x07AF	<i>CurrentTier10 Block16Summation Delivered</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O

0x07B0	<i>CurrentTier11 Block1Summation Delivered</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x07BN	<i>CurrentTier11 BlockN+1 SummationDelivered</i> ...	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x07BF	<i>CurrentTier11 Block16Summation Delivered</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x07C0	<i>CurrentTier12 Block1Summation Delivered</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x07CN	<i>CurrentTier12 BlockN+1 SummationDelivered</i> ...	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x07CF	<i>CurrentTier12 Block16Summation Delivered</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x07D0	<i>CurrentTier13 Block1Summation Delivered</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x07DN	<i>CurrentTier13 BlockN+1 SummationDelivered</i> ...	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x07DF	<i>CurrentTier13 Block16Summation Delivered</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x07E0	<i>CurrentTier14 Block1Summation Delivered</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x07EN	<i>CurrentTier14 BlockN+1 SummationDelivered</i> ...	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x07EF	<i>CurrentTier14 Block16Summation Delivered</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x07F0	<i>CurrentTier15 Block1Summation Delivered</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O

0x07FN	<i>CurrenTier15 BlockN+1 SummationDelivered</i> ...	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x07FF	<i>CurrentTier15 Block16Summation Delivered</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O

6272

6273 D.3.2.2.8.1 CurrentTierNBlockNSummationDelivered Attributes

6274 Attributes *CurrentNoTierBlock1SummationDelivered* through *CurrentTier15Block16SummationDelivered* represent the most recent summed value of Energy, Gas, or Water delivered to the premises (i.e delivered to the customer from the utility) at a specific price tier as defined by a TOU schedule, Block Threshold or a real time pricing period. If optionally provided, attributes *CurrentNoTierBlock1SummationDelivered* through *CurrentTier15Block16SummationDelivered* are updated continuously as new measurements are made.

6281 **Note:** SummationFormatting shall be used against the Block Information attribute set. The expected practical limit for the number of Block attributes supported is 64. The CurrentTierNBlockNSummationDelivered attributes are reset at the start of each Block 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 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

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	0x0000000000000000 - 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

6306 may have a different meaning. See sub-clause D.3.2.2.3. In the case of overlap of alarm codes
 6307 from the Generic Alarm Group with codes in other groups, e.g. Burst Detect, it is
 6308 recommended to only use the code of the Generic Alarm Group.

6309

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

6310

6311 The Electricity Alarm Group defines alarms specific for electricity meters as defined below.

6312

Table D-36– Electricity Alarm Group

Alarm Code	Alarm Condition
10	Low Voltage L1
11	High Voltage L1

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

36	Flow sensor air detect
37	Pipe empty
38-3F	Reserved

- 6317
 6318 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

- 6320
 6321 The Heat and Cooling Specific Alarm Group defines alarms specific for Heat or Cooling meters
 6322 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

- 6324
 6325 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

- 6327
 6328 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

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 below, these are used for meter specific functionality that is not covered by the current smart energy specification.
 6332
 6333

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

D.3.2.2.10 Block Information Attribute Set (Received)

The following set of attributes provides a remote access to the Electric, Gas, or Water metering devices block readings. The Block Information attribute set supports Block pricing and combined Tier-Block pricing, the number of blocks is one greater than the number of block thresholds defined in the Pricing cluster.

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	<i>CurrentNoTier Block1Summation Received</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0901	<i>CurrentNoTier Block2Summation Received</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0902	<i>CurrentNoTier Block3Summation Received</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
...
0x090F	<i>CurrentNoTier Block16Summation Received</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O

0x0910	<i>CurrentTier1 Block1Summation Received</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0911	<i>CurrentTier1 Block2Summation Received</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0912	<i>CurrentTier1 Block3Summation Received</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
...
0x091F	<i>CurrentTier1 Block16Summation Received</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0920	<i>CurrentTier2 Block1Summation Received</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
...
0x092F	<i>CurrentTier2 Block16Summation Received</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0930	<i>CurrentTier3 Block1Summation Received</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
...
0x093F	<i>CurrentTier3 Block16Summation Received</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0940	<i>CurrentTier4 Block1Summation Received</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
...
0x094F	<i>CurrentTier4 Block16 SummationReceived</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0950	<i>CurrentTier5 Block1Summation Received</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
...
0x095F	<i>CurrentTier5 Block16Summation Received</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O

0x0960	<i>CurrentTier6 Block1Summation Received</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
...
0x096F	<i>CurrentTier6 Block16Summation Received</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0970	<i>CurrentTier7 Block1Summation Received</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
...
0x097F	<i>CurrentTier7 Block16Summation Received</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0980	<i>CurrentTier8 Block1Summation Received</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
...
0x098F	<i>CurrentTier8 Block16Summation Received</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x0990	<i>CurrentTier9 Block1Summation Received</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
...
0x099F	<i>CurrentTier9 Block16Summation Received</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x09A0	<i>CurrentTier10 Block1Summation Received</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
...
0x09AF	<i>CurrentTier10 Block16Summation Received</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x09B0	<i>CurrentTier11 Block1Summation Received</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
...

0x09BF	<i>CurrentTier11 Block16Summation Received</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x09C0	<i>CurrentTier12 Block1Summation Received</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
...
0x09CF	<i>CurrentTier12 Block16Summation Received</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x09D0	<i>CurrentTier13 Block1Summation Received</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
...
0x09DF	<i>CurrentTier13 Block16Summation Received</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x09E0	<i>CurrentTier14 Block1Summation Received</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
...
0x09EF	<i>CurrentTier14 Block16Summation Received</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x09F0	<i>CurrentTier15 Block1Summation Received</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
...
0x09FF	<i>CurrentTier15 Block16Summation Received</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O

6343

6344 D.3.2.2.10.1 CurrentTierNBlockNSummationReceived Attributes

6345 Attributes *CurrentNoTierBlock1SummationReceived* through
 6346 *CurrentTier15Block16SummationReceived* represent the most recent summed value of Energy,
 6347 Gas, or Water received from the premises (i.e. delivered to the utility from the customer) at a
 6348 specific price tier as defined by a TOU schedule, Block Threshold or a real time pricing period. If
 6349 optionally provided, attributes *CurrentNoTierBlock1SummationReceived* through
 6350 *CurrentTier15Block16SummationReceived* are updated continuously as new measurements are
 6351 made.

6352 **Note:** *SummationFormatting* shall be used against the Block Information attribute set. The
6353 practical limit for the number of Block attributes supported is 32. The
6354 *CurrentTierNBlockNSummationReceived* attributes are reset at the start of each Block Threshold
6355 Period.

6356 D.3.2.2.11 Meter Billing Attribute Set

6357 The billing information within this attribute set is created on the metering device. The
6358 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					

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.

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 Set6401 **Table D-45– Supply Control Attribute Set**

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0B00	ProposedChangeSupplyImplementationTime	UTC Time		Read only	-	O
0x0B01	ProposedChangeSupplyStatus	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.

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 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 is
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

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	<i>AlternativeInstantaneousDemand</i>	Signed 24-bit Integer	-8,388,607 to 8,388,607	Read Only	0x00	O
0x0C01	<i>CurrentDayAlternative ConsumptionDelivered</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFFF	Read Only	-	O
0x0C02	<i>CurrentDayAlternative ConsumptionReceived</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFFF	Read Only	-	O
0x0C03	<i>PreviousDayAlternative ConsumptionDelivered</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFFF	Read Only	-	O
0x0C04	<i>PreviousDayAlternative ConsumptionReceived</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFFF	Read Only	-	O
0x0C05	<i>CurrentAlternative PartialProfileInterval StartTimeDelivered</i>	UTCTime		Read Only	-	O
0x0C06	<i>CurrentAlternative PartialProfileInterval StartTimeReceived</i>	UTCTime	0x000000 to 0xFFFFFFFF	Read Only	-	O
0x0C07	<i>CurrentAlternative PartialProfileInterval ValueDelivered</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFFF	Read Only	-	O
0x0C08	<i>CurrentAlternative PartialProfileInterval ValueReceived</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFFF	Read Only	-	O
0x0C09	<i>CurrentDay AlternativeMaxPressure</i>	Unsigned 48-bit Integer	0x000000 000000 to 0xFFFFFFFF FFFFFFFF	Read Only	-	O
0x0C0A	<i>CurrentDay AlternativeMinPressure</i>	Unsigned 48-bit Integer	0x000000 000000 to 0xFFFFFFFF FFFFFFFF	Read Only	-	O
0x0C0B	<i>PreviousDay AlternativeMaxPressure</i>	Unsigned 48-bit Integer	0x000000 000000 to 0xFFFFFFFF FFFFFFFF	Read Only	-	O

0x0C0C	<i>PreviousDay AlternativeMinPressure</i>	Unsigned 48-bit Integer	0x000000 000000 to 0xFFFFFFF FFFFFFF	Read Only	-	O
0x0C0D	<i>CurrentDay AlternativeMaxDemand</i>	Signed 24-bit Integer	-8,388,607 to 8,388,607	Read Only	-	O
0x0C0E	<i>PreviousDay AlternativeMaxDemand</i>	Signed 24-bit Integer	-8,388,607 to 8,388,607	Read Only	-	O
0x0C0F	<i>CurrentMonth AlternativeMaxDemand</i>	Signed 24-bit Integer	-8,388,607 to 8,388,607	Read Only	-	O
0x0C10	<i>CurrentYear AlternativeMaxDemand</i>	Signed 24-bit Integer	-8,388,607 to 8,388,607	Read Only	-	O
0x0C11 to 0x0C1F	Reserved					
0x0C20	<i>PreviousDay2Alternative ConsumptionDelivered</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0C21	<i>PreviousDay2Alternative ConsumptionReceived</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0C22	<i>PreviousDay3Alternative ConsumptionDelivered</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0C23	<i>PreviousDay3Alternative ConsumptionReceived</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0C24	<i>PreviousDay4Alternative ConsumptionDelivered</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0C25	<i>PreviousDay4Alternative ConsumptionReceived</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0C26	<i>PreviousDay5Alternative ConsumptionDelivered</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0C27	<i>PreviousDay5Alternative ConsumptionReceived</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0C28	<i>PreviousDay6Alternative ConsumptionDelivered</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0C29	<i>PreviousDay6Alternative ConsumptionReceived</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O

0x0C2A	<i>PreviousDay7Alternative ConsumptionDelivered</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0C2B	<i>PreviousDay7Alternative ConsumptionReceived</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0C2C	<i>PreviousDay8Alternative ConsumptionDelivered</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0C2D	<i>PreviousDay8Alternative ConsumptionReceived</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0C2E to 0x0C2F	Reserved					
0x0C30	<i>CurrentWeekAlternative ConsumptionDelivered</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0C31	<i>CurrentWeekAlternative ConsumptionReceived</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0C32	<i>PreviousWeekAlternative ConsumptionDelivered</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0C33	<i>PreviousWeekAlternative ConsumptionReceived</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0C34	<i>PreviousWeek2Alternative ConsumptionDelivered</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0C35	<i>PreviousWeek2Alternative ConsumptionReceived</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0C36	<i>PreviousWeek3Alternative ConsumptionDelivered</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0C37	<i>PreviousWeek3Alternative ConsumptionReceived</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0C38	<i>PreviousWeek4Alternative ConsumptionDelivered</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0C39	<i>PreviousWeek4Alternative ConsumptionReceived</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O

0x0C3A	<i>PreviousWeek5Alternative ConsumptionDelivered</i>	Unsigned 24-bit Integer	0x0000000 to 0xFFFFFFF	Read Only	-	O
0x0C3B	<i>PreviousWeek5Alternative ConsumptionReceived</i>	Unsigned 24-bit Integer	0x0000000 to 0xFFFFFFF	Read Only	-	O
0x0C3C to 0x0C3F	Reserved					
0x0C40	<i>CurrentMonthAlternative ConsumptionDelivered</i>	Unsigned 32-bit Integer	0x000000000 to 0xFFFFFFFF	Read Only	-	O
0x0C41	<i>CurrentMonthAlternative ConsumptionReceived</i>	Unsigned 32-bit Integer	0x000000000 to 0xFFFFFFFF	Read Only	-	O
0x0C42	<i>PreviousMonthAlternative ConsumptionDelivered</i>	Unsigned 32-bit Integer	0x000000000 to 0xFFFFFFFF	Read Only	-	O
0x0C43	<i>PreviousMonthAlternative ConsumptionReceived</i>	Unsigned 32-bit Integer	0x000000000 to 0xFFFFFFFF	Read Only	-	O
0x0C44	<i>PreviousMonth2Alternative ConsumptionDelivered</i>	Unsigned 32-bit Integer	0x000000000 to 0xFFFFFFFF	Read Only	-	O
0x0C45	<i>PreviousMonth2Alternative ConsumptionReceived</i>	Unsigned 32-bit Integer	0x000000000 to 0xFFFFFFFF	Read Only	-	O
0x0C46	<i>PreviousMonth3Alternative ConsumptionDelivered</i>	Unsigned 32-bit Integer	0x000000000 to 0xFFFFFFFF	Read Only	-	O
0x0C47	<i>PreviousMonth3Alternative ConsumptionReceived</i>	Unsigned 32-bit Integer	0x000000000 to 0xFFFFFFFF	Read Only	-	O
0x0C48	<i>PreviousMonth4Alternative ConsumptionDelivered</i>	Unsigned 32-bit Integer	0x000000000 to 0xFFFFFFFF	Read Only	-	O
0x0C49	<i>PreviousMonth4Alternative ConsumptionReceived</i>	Unsigned 32-bit Integer	0x000000000 to 0xFFFFFFFF	Read Only	-	O
0x0C4A	<i>PreviousMonth5Alternative ConsumptionDelivered</i>	Unsigned 32-bit Integer	0x000000000 to 0xFFFFFFFF	Read Only	-	O
0x0C4B	<i>PreviousMonth5Alternative ConsumptionReceived</i>	Unsigned 32-bit Integer	0x000000000 to 0xFFFFFFFF	Read Only	-	O
0x0C4C	<i>PreviousMonth6Alternative ConsumptionDelivered</i>	Unsigned 32-bit Integer	0x000000000 to 0xFFFFFFFF	Read Only	-	O

0x0C4D	<i>PreviousMonth6Alternative ConsumptionReceived</i>	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0C4E	<i>PreviousMonth7Alternative ConsumptionDelivered</i>	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0C4F	<i>PreviousMonth7Alternative ConsumptionReceived</i>	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0C50	<i>PreviousMonth8 Alternative ConsumptionDelivered</i>	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0C51	<i>PreviousMonth8Alternative ConsumptionReceived</i>	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0C52	<i>PreviousMonth9Alternative ConsumptionDelivered</i>	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0C53	<i>PreviousMonth9Alternative ConsumptionReceived</i>	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0C54	<i>PreviousMonth10Alternative ConsumptionDelivered</i>	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0C55	<i>PreviousMonth10 Alternative ConsumptionReceived</i>	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0C56	<i>PreviousMonth11 Alternative ConsumptionDelivered</i>	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0C57	<i>PreviousMonth11Alternative ConsumptionReceived</i>	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0C58	<i>PreviousMonth12Alternative ConsumptionDelivered</i>	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0C59	<i>PreviousMonth12Alternative ConsumptionReceived</i>	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0C5A	<i>PreviousMonth13Alternative ConsumptionDelivered</i>	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0C5B	<i>PreviousMonth13Alternative ConsumptionReceived</i>	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0C5C to 0x0CFF	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 *Alternative 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.

6476 D.3.2.2.13.8 CurrentAlternativePartialProfileIntervalValueDelivered Attribute

6477 *CurrentAlternativePartialProfileIntervalValueDelivered* represents the value of the current
6478 Load Profile interval being accumulated for commodity delivered.

6479 D.3.2.2.13.9 CurrentAlternativePartialProfileIntervalValueReceived Attribute

6480 *CurrentAlternativePartialProfileIntervalValueReceived* represents the value of the current
6481 Load Profile interval being accumulated for commodity received.

6482 D.3.2.2.13.10 CurrentDayAlternativeMaxPressure Attribute

6483 *CurrentDayAlternativeMaxPressure* is the maximum pressure reported during a day from the
6484 water or gas meter.

6485 D.3.2.2.13.11 PreviousDayAlternativeMaxPressure Attribute

6486 *PreviousDayAlternativeMaxPressure* represents the maximum pressure reported during previous
6487 day from the water or gas meter.

6488 D.3.2.2.13.12 CurrentDayAlternativeMinPressure Attribute

6489 *CurrentDayAlternativeMinPressure* is the minimum pressure reported during a day from the
6490 water or gas meter.

6491 D.3.2.2.13.13 PreviousDayAlternativeMinPressure Attribute

6492 *PreviousDayAlternativeMinPressure* represents the minimum pressure reported during previous
6493 day from the water or gas meter.

6494 D.3.2.2.13.14 CurrentDayAlternativeMaxDemand Attribute

6495 *CurrentDayAlternativeMaxDemand* represents the maximum demand or rate of delivered
6496 value of Energy, Gas, or Water being utilized at the premises.

6497 D.3.2.2.13.15 PreviousDayAlternativeMaxDemand Attribute

6498 *PreviousDayAlternativeMaxDemand* represents the maximum demand or rate of delivered
6499 value of Energy, Gas, or Water being utilized at the premises.

6500 **Note:** At the end of a day the metering device will transfer the
6501 *CurrentDayAlternativeMaxPressure* into *PreviousDayAlternativeMaxPressure*,
6502 *CurrentDayAlternativeMinPressure* into *PreviousDayAlternativeMinPressure* and
6503 *CurrentDayAlternativeMaxDemand* into *PreviousDayAlternativeMaxDemand*.

6504 D.3.2.2.13.16 CurrentMonthAlternativeMaxDemand Attribute

6505 *CurrentMonthAlternativeMaxDemand* is the maximum demand reported during a month from
6506 the meter.

- 6507 For electricity, heat and cooling meters this is the maximum power reported in a month.
- 6508 D.3.2.2.13.17 CurrentYearAlternativeMaxDemand Attribute
- 6509 *CurrentYearAlternativeMaxDemand* is the maximum demand reported during a year from the meter.
- 6511 For electricity, heat and cooling meters this is the maximum power reported in a year.
- 6512 D.3.2.2.13.18 PreviousDayNAlternativeConsumptionDelivered Attribute
- 6513 *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.
- 6516 D.3.2.2.13.19 PreviousDayNAlternativeConsumptionReceived Attribute
- 6517 *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.
- 6520 D.3.2.2.13.20 CurrentWeekAlternativeConsumptionDelivered Attribute
- 6521 *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.
- 6526 D.3.2.2.13.21 CurrentWeekAlternativeConsumptionReceived Attribute
- 6527 *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.
- 6532 D.3.2.2.13.22 PreviousWeekNAlternativeConsumptionDelivered Attribute
- 6533 *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.
- 6537 D.3.2.2.13.23 PreviousWeekNAlternativeConsumptionReceived Attribute
- 6538 *PreviousWeekNAlternativeConsumptionReceived* represents the summed value received from the premises within the previous week period starting at the *Historical Freeze Time* (HFT) on the

6540 Monday to the Sunday. If the optional HFT attribute is not available, default to midnight local
 6541 time.

6542 D.3.2.2.13.24 CurrentMonthAlternativeConsumptionDelivered Attribute

6543 *CurrentMonthAlternativeConsumptionDelivered* represents the summed value delivered to the
 6544 premises since the *Historical Freeze Time* (HFT) on the 1st of the month to the last HFT read. If
 6545 optionally provided, *CurrentMonthAlternativeConsumptionDelivered* is updated continuously as
 6546 new measurements are made. If the optional HFT attribute is not available, default to midnight
 6547 local time.

6548 D.3.2.2.13.25 CurrentMonthAlternativeConsumptionReceived Attribute

6549 *CurrentMonthAlternativeConsumptionReceived* represents the summed value received from the
 6550 premises since the *Historical Freeze Time* (HFT) on the 1st of the month to the last HFT read. If
 6551 optionally provided, *CurrentMonthAlternativeConsumptionReceived* is updated continuously as
 6552 new measurements are made. If the optional HFT attribute is not available, default to midnight
 6553 local time.

6554 D.3.2.2.13.26 PreviousMonthNAlternativeConsumptionDelivered Attribute

6555 *PreviousMonthNAlternativeConsumptionDelivered* represents the summed value delivered to the
 6556 premises within the previous Month period starting at the *Historical Freeze Time* (HFT) on the
 6557 1st of the month to the last day of the month. If the optional HFT attribute is not available, default
 6558 to midnight local time.

6559 D.3.2.2.13.27 PreviousMonthNAlternativeConsumptionReceived Attribute

6560 *PreviousMonthNAlternativeConsumptionReceived* represents the summed value received from
 6561 the premises within the previous month period starting at the *Historical Freeze Time* (HFT) on
 6562 the 1st of the month to the last day of the month. If the optional HFT attribute is not available,
 6563 default to midnight local time.

6564

6565 D.3.2.3 Server Commands

6566 D.3.2.3.1 Commands Generated

6567 The command IDs generated by the Metering server cluster are listed in Table D-47.

6568

Table D-47– Generated Command IDs for the Metering Server

Command Identifier Field Value	Description	Mandatory / Optional
0x00	<i>Get Profile Response</i>	O

0x01	<i>Request Mirror</i>	O
0x02	<i>Remove Mirror</i>	O
0x03	<i>Request Fast Poll Mode Response</i>	O
0x04	<i>ScheduleSnapshot Response</i>	O
0x05	<i>TakeSnapshotResponse</i>	O
0x06	<i>Publish Snapshot</i>	O
0x07	<i>GetSampledData Response</i>	O
0x08	<i>ConfigureMirror</i>	O
0x09	<i>ConfigureNotification Scheme</i>	O
0x0A	<i>ConfigureNotification Flag</i>	O
0x0B	<i>GetNotifiedMessage</i>	O
0x0C	<i>Supply Status Response</i>	O
0x0D	<i>StartSamplingResponse</i>	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

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

6591 calculated when the packets are received and can replace the usage of this field. The intent is to
6592 provide this information as a convenience.

6593 **Intervals:** Series of interval data captured using the period specified by the ProfileIntervalPeriod
6594 field. The content of the interval data depends of the type of information requested using the
6595 Channel field in the *Get Profile* Command, and will represent the change in that information
6596 since the previous interval. Data is organized in a reverse chronological order, the most
6597 recent interval is transmitted first and the oldest interval is transmitted last. Invalid intervals
6598 should be marked as 0xFFFF.

6599 **D.3.2.3.1.1.3 When Generated**

6600 This command is generated when the Client command *GetProfile* is received. Please refer to
6601 sub-clause D.3.3.3.1.1.

6602

6603 **D.3.2.3.1.2 Request Mirror Command**

6604 This command is used to request the ESI to mirror Metering Device data.

6605 **D.3.2.3.1.2.1 Payload Details**

6606 There are no fields for this command.

6607 **D.3.2.3.1.2.2 Effect on Receipt**

6608 On receipt of this command, the Server shall send a *RequestMirrorReponse* command (see
6609 sub-clause D.3.3.3.1.2).

6610

6611 **D.3.2.3.1.3 Remove Mirror Command**

6612 This command is used to request the ESI to remove its mirror of Metering Device data. The
6613 device sending the *Remove Mirror* command to the ESI shall send the command to the mirror
6614 endpoint to be removed. Only the device that created the mirror on the ESI or the ESI itself
6615 should be allowed to remove the mirror from the ESI.

6616 **D.3.2.3.1.3.1 Payload Details**

6617 There are no fields for this command.

6618 **D.3.2.3.1.3.2 Effect on Receipt**

6619 On receipt of this command, the Server shall send a *MirrorRemoved* command (see sub-clause
6620 D.3.3.3.1.3).

6621

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.

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 value contained in this field indicates the value allocated to the *ScheduleSnapshot* command for 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

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**

6684 **D.3.2.3.1.7.2 Payload Details**

6685 **Snapshot ID (mandatory):** Unique identifier allocated by the device creating the snapshot.

6686 **Snapshot Time (mandatory):** This is a 32 bit value (in UTC Time) representing the time at
6687 which the data snapshot was taken.

6688 **Total Snapshots Found (mandatory):** An 8-bit Integer indicating the number of snapshots
6689 found, based on the search criteria defined in the associated *GetSnapshot* command. If the value
6690 is greater than 1, the client is able to request the next snapshot by incrementing the *Snapshot*
6691 *Offset* field in an otherwise repeated *GetSnapshot* command.

6692 **Command Index (mandatory):** The *CommandIndex* is used to count the payload fragments in
6693 the case where the entire payload (snapshot) does not fit into one message. The *CommandIndex*
6694 starts at 0 and is incremented for each fragment belonging to the same command.

6695 **Total Number of Commands (mandatory):** In the case where the entire payload (snapshot)
6696 does not fit into one message, the *Total Number of Commands* field indicates the total number of
6697 sub-commands that will be returned.

6698 **Snapshot Cause (mandatory):** A 32-bit BitMap indicating the cause of the snapshot. The
6699 snapshot cause values are listed in Table D-52.

6700

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

Bit	Cause Description
19	Reserved for Prepayment cluster
20 – 31	Reserved

6701 **SnapshotPayloadType (mandatory):** The *SnapshotPayloadType* is an 8-bit enumerator
 6702 defining the format of the *SnapshotSub-Payload* in this message. The different snapshot types are
 6703 listed in Table D-53. The server selects the *SnapshotPayloadType* based on the charging scheme
 6704 in use.

6705 **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

6706 If the snapshot is taken by the server due to a change of Tariff Information (cause = 3) which
 6707 involves a change in charging scheme then two snapshots shall be taken, the first according to the
 6708 charging scheme being dismissed, the second to the scheme being introduced.

6709 **SnapshotSub-Payload (mandatory):** the format of the *SnapshotSub-Payload* differs depending
 6710 on the *SnapshotPayloadType*, as shown below. Note that, where the entire payload (snapshot)
 6711 does not fit into one message, only the leading (non-Sub-Payload) fields of the Snapshot payload
 6712 are repeated in each command; the *SnapshotSub-Payload* is divided over the required number of
 6713 commands. Figure D-18 explains this further.

Snapshot as a single command, exceeds the maximum payload size

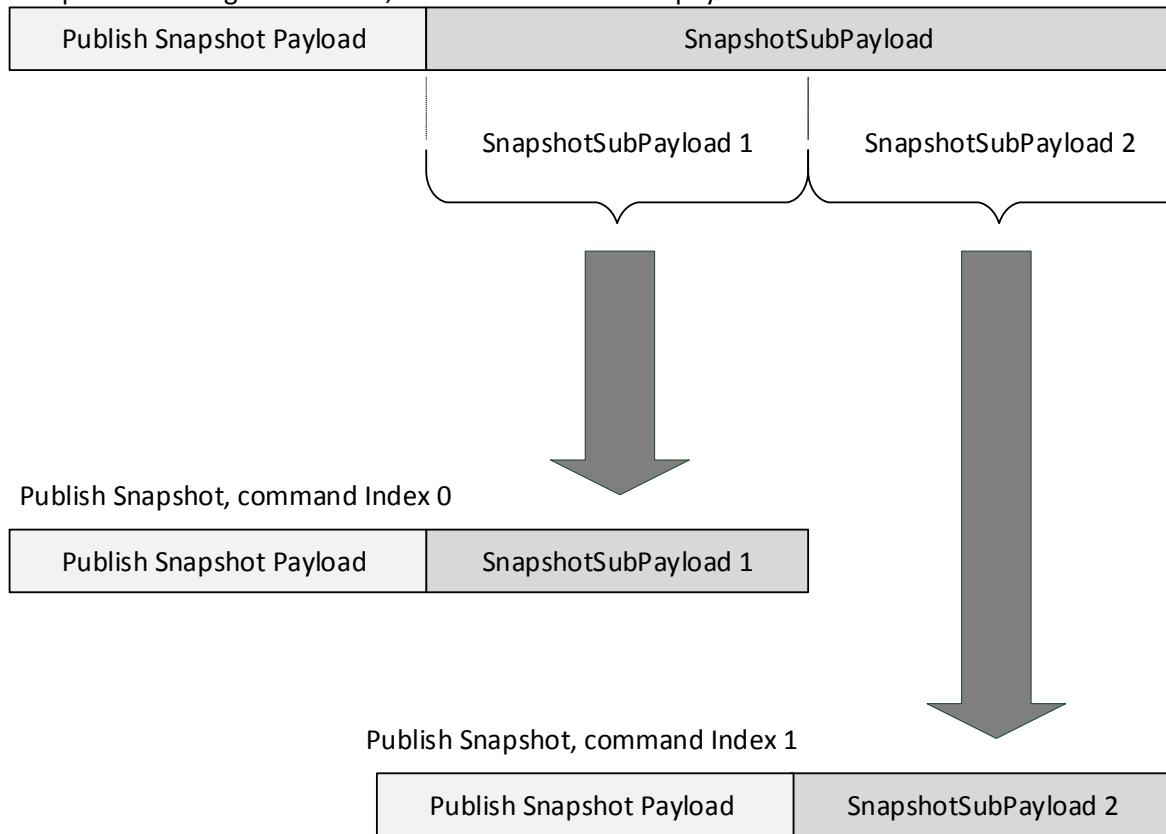


Figure D-18– Snapshot Utilizing Multiple Commands

6714
6715

6716

6717 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)

6718 **Figure D-19– TOU Information Delivered Snapshot Sub-Payload**

6719 **Current Summation Delivered (mandatory):** An unsigned 48-bit integer that returns the value of the *CurrentSummationDelivered* attribute at the stated snapshot timestamp.

6720

6721 **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.

6722

6723

6724 **BillToDateTimeStampDelivered (mandatory):** A UTC timestamp that indicates when the value of the associated *BillToDateDelivered* parameter was last updated.

6725

6726 **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.

6727

6728

6729

6730 **ProjectedBillTimeStampDelivered (mandatory):** A UTC timestamp that indicates when the associated *ProjectedBillDelivered* parameter was last updated.

6731

6732 **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.

6733

6734

6735

6736 **Number of Tiers in Use (mandatory):** An 8-bit integer representing the number of tiers in use at the time the snapshot was taken.

6737

6738 **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.

6744 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)	BillToDateTimeStamp Received (M)	Projected Bill Received (M)	ProjectedBillTimeStamp Received (M)	Bill Received Trailing Digit (M)	Number of Tiers in Use (M)	Tier Summation (M)

6745 **Figure D-20– TOU Information Received Snapshot Sub-Payload**

6746 **Current Summation Received (mandatory):** An unsigned 48-bit integer that returns the
6747 value of the *CurrentSummationReceived* attribute at the stated snapshot timestamp. A value of
6748 0xFFFFFFFFFFFF means not available.

6749 **BillToDateReceived (mandatory):** An unsigned 32-bit integer that provides a value for the
6750 costs in the current billing period. This value is measured in a base unit of Currency with the
6751 decimal point located as indicated by the *BillReceivedTrailingDigit* field.

6752 **BillToDateTimeStampReceived (mandatory):** A UTC timestamp that indicates when the
6753 value of the associated *BillToDateReceived* parameter was last updated.

6754 **ProjectedBillReceived (mandatory):** An unsigned 32-bit integer that provides a value
6755 indicating what the estimated state of the account will be at the end of the billing period based
6756 on past generation. This attribute is measured in a base unit of Currency with the decimal
6757 point located as indicated by the *BillReceivedTrailingDigit* field.

6758 **ProjectedBillTimeStampReceived (mandatory):** A UTC timestamp that indicates when the
6759 associated *ProjectedBillReceived* parameter was last updated.

6760 **BillReceivedTrailingDigit (mandatory):** An 8-bit BitMap used to determine where the
6761 decimal point is located in the *BillToDateReceived* and *ProjectedBillReceived* fields. The
6762 most significant nibble indicates the number of digits to the right of the decimal point. The
6763 least significant nibble is reserved and shall be 0.

6764 **Number of Tiers in Use (mandatory):** An 8-bit integer representing the number of tiers in
6765 use at the time the snapshot was taken.

6766 **TierSummation (mandatory):** The *Publish Snapshot* command contains N elements of
6767 *CurrentTierNSummationReceived* attributes from the TOU Information Set. The Metering
6768 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.

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)	BillToDateTimeStamp Delivered (M)	Projected Bill Delivered (M)	ProjectedBillTimeStamp 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.

6792 **Number of Tiers in Use (mandatory):** An 8-bit integer representing the number of tiers in
6793 use at the time the snapshot was taken.

6794 **TierSummation (mandatory):** The *Publish Snapshot* command contains N elements of
6795 *CurrentTierNSummationDelivered* attributes from the TOU Information Set. The Metering
6796 server shall send only the number of tiers in use, as stated in this command. The first element
6797 of the TOU Information Set (Delivered), is *CurrentTier1Summation*. For the following
6798 elements, the tier index is incremented until the number of tiers in use is reached.

6799 **Number of Tiers and Block Thresholds in Use (mandatory):** An 8-bit BitMap representing
6800 the number of tiers and block thresholds in use at the time the snapshot was taken. The most
6801 significant nibble defines the number of tiers in use, whereas the least significant nibble
6802 indicates the number of block thresholds in use.

6803 **TierBlockSummation (T,B):** The *Publish Snapshot* command contains N elements of the
6804 Block Information Attribute Set (Delivered). The metering server shall send only the number
6805 of Tiers and Blocks in use as stated in this command. The Block Information Attribute Set has
6806 two dimensions, the row – tier index (T) and the block – column index (B).

6807 The first element of the Tier Block Summation field is
6808 *CurrentTier1Block1SummationDelivered* attribute. For the following elements, the block
6809 index is incremented until the number of blocks in use is reached. Then the tier index is
6810 incremented and the block index starts at 1 again. This continues until the stated number of
6811 tiers in use is reached.

6812 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

6815 **Current Summation Received (mandatory):** An unsigned 48-bit integer that returns the
6816 value of the *CurrentSummationReceived* attribute at the stated snapshot timestamp. A value of
6817 0xFFFFFFFFFFFF means not available.

6818 **BillToDateReceived (mandatory):** An unsigned 32-bit integer that provides a value for the
6819 costs in the current billing period. This value is measured in a base unit of Currency with the
6820 decimal point located as indicated by the *BillReceivedTrailingDigit* field.

6821 **BillToDateTimeStampReceived (mandatory):** A UTC timestamp that indicates when the
6822 value of the associated *BillToDateReceived* parameter was last updated.

6823 **ProjectedBillReceived (mandatory):** An unsigned 32-bit integer that provides a value
6824 indicating what the estimated state of the account will be at the end of the billing period based
6825 on past generation. This attribute is measured in a base unit of Currency with the decimal
6826 point located as indicated by the *BillReceivedTrailingDigit* field.

6827 **ProjectedBillTimeStampReceived (mandatory):** A UTC timestamp that indicates when the
6828 associated *ProjectedBillReceived* parameter was last updated.

6829 **BillReceivedTrailingDigit (mandatory):** An 8-bit BitMap used to determine where the
6830 decimal point is located in the *BillToDateReceived* and *ProjectedBillReceived* fields. The
6831 most significant nibble indicates the number of digits to the right of the decimal point. The
6832 least significant nibble is reserved and shall be 0.

6833 **Number of Tiers in Use (mandatory):** An 8-bit integer representing the number of tiers in
6834 use at the time the snapshot was taken.

6835 **TierSummation (mandatory):** The *Publish Snapshot* command contains N elements of
6836 *CurrentTierNSummationReceived* attributes from the TOU Information Set. The Metering
6837 server shall send only the number of tiers in use, as stated in this command. The first element
6838 of the TOU Information Set (Received), is *CurrentTier1Summation*. For the following
6839 elements, the tier index is incremented until the number of tiers in use is reached.

6840 **Number of Tiers and Block Thresholds in Use (mandatory):** An 8-bit BitMap representing
6841 the number of tiers and block thresholds in use at the time the snapshot was taken. The most
6842 significant nibble defines the number of tiers in use, whereas the least significant nibble
6843 indicates the number of block thresholds in use.

6844 **TierBlockSummation (T,B):** The *Publish Snapshot* command contains N elements of the
6845 Block Information Attribute Set (Received). The metering server shall send only the number
6846 of Tiers and Blocks in use as stated in this command. The Block Information Attribute Set has
6847 two dimensions, the row – tier index (T) and the block – column index (B).

6848 The first element of the Tier Block Summation field is
 6849 *CurrentTier1Block1SummationReceived* attribute. For the following elements, the block index
 6850 is incremented until the number of blocks in use is reached. Then the tier index is incremented
 6851 and the block index starts at 1 again. This continues until the stated number of tiers in use is
 6852 reached.

6853 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)

6854 **Figure D-23– TOU Information Delivered (No Billing) Snapshot Sub-Payload**

6855 **Current Summation Delivered (mandatory):** An unsigned 48-bit integer that returns the
 6856 value of the *CurrentSummationDelivered* attribute at the stated snapshot timestamp.

6857 **Number of Tiers in Use (mandatory):** An 8-bit integer representing the number of tiers in
 6858 use at the time the snapshot was taken.

6859 **TierSummation (mandatory):** The *Publish Snapshot* command contains N elements of
 6860 *CurrentTierNSummationDelivered* attributes from the TOU Information Set. The Metering
 6861 server shall send only the number of tiers in use, as stated in this command. The first element
 6862 of the TOU Information Set (Delivered), is *CurrentTier1Summation*. For the following
 6863 elements, the tier index is incremented until the number of tiers in use is reached.

6864 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)

6866 **Figure D-24– TOU 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.

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

6895 of Tiers and Blocks in use as stated in this command. The Block Information Attribute Set has
6896 two dimensions, the row – tier index (T) and the block – column index (B).

6897 The first element of the Tier Block Summation field is
6898 *CurrentTier1Block1SummationDelivered* attribute. For the following elements, the block
6899 index is incremented until the number of blocks in use is reached. Then the tier index is
6900 incremented and the block index starts at 1 again. This continues until the stated number of
6901 tiers in use is reached.

6902 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)

6903 **Figure D-26– Block Information Received (No Billing) Snapshot Sub-Payload**

6904 **Current Summation Received (mandatory):** An unsigned 48-bit integer that returns the
6905 value of the *CurrentSummationReceived* attribute at the stated snapshot timestamp. A value of
6906 0xFFFFFFFFFFFF means not available.

6907 **Number of Tiers in Use (mandatory):** An 8-bit integer representing the number of tiers in
6908 use at the time the snapshot was taken.

6909 **TierSummation (mandatory):** The *Publish Snapshot* command contains N elements of
6910 *CurrentTierNSummationReceived* attributes from the TOU Information Set. The Metering
6911 server shall send only the number of tiers in use, as stated in this command. The first element
6912 of the TOU Information Set (Received), is *CurrentTier1Summation*. For the following
6913 elements, the tier index is incremented until the number of tiers in use is reached.

6914 **Number of Tiers and Block Thresholds in Use (mandatory):** An 8-bit BitMap representing
6915 the number of tiers and block thresholds in use at the time the snapshot was taken. The most
6916 significant nibble defines the number of tiers in use, whereas the least significant nibble
6917 indicates the number of block thresholds in use.

6918 **TierBlockSummation (T,B):** The *Publish Snapshot* command contains N elements of the
6919 Block Information Attribute Set (Received). The metering server shall send only the number
6920 of Tiers and Blocks in use as stated in this command. The Block Information Attribute Set has
6921 two dimensions, the row – tier index (T) and the block – column index (B).

6922 The first element of the Tier Block Summation field is
6923 *CurrentTier1Block1SummationReceived* attribute. For the following elements, the block index

6924 is incremented until the number of blocks in use is reached. Then the tier index is incremented
 6925 and the block index starts at 1 again. This continues until the stated number of tiers in use is
 6926 reached.

6927 **D.3.2.3.1.7.3 When Generated**

6928 A *Publish Snapshot* command is generated in response to *GetSnapshot* command or when a new
 6929 snapshot is created. The device shall send a single *Publish Snapshot* command according to the
 6930 search criteria defined in the associated *GetSnapshot* command. A ZCL Default Response with
 6931 status NOT_FOUND shall be returned if there is no appropriate snapshot data available.

6932 **D.3.2.3.1.8 GetSampledDataResponse Command**

6933 This command is used to send the requested sample data to the client. It is generated in response
 6934 to a *GetSampledData* command (see D.3.3.3.1.9).

6935 **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)	SampleStartTime (M)	SampleType (M)	SampleRequest Interval (M)	NumberOfSamples (M)	Samples (M)

6936 Figure D-27– Format of the *GetSampledDataResponse* Command Payload

6937 **D.3.2.3.1.8.2 Payload Details**

6938 **SampleID (mandatory):** Unique identifier allocated to this Sampling session. This field allows
 6939 devices to match response data with the appropriate request. See D.3.2.3.1.14 for further details.

6940 **SampleStartTime (mandatory):** A UTC Time field to denote the time of the first sample
 6941 returned in this response.

6942 **SampleType (mandatory):** An 8 bit enumeration that identifies the type of data being sampled.
 6943 Possible values are defined in the following table:

6944 **Table D-54– Sample Type Enumerations**

Enumeration	Description
0	Consumption Delivered
1 - 255	Reserved

6945

6946 **SampleRequestInterval (mandatory):** An unsigned 16-bit field representing the interval or
6947 time in seconds between samples.

6948 **NumberOfSamples (mandatory):** Represents the number of samples being requested. This
6949 value cannot exceed the size stipulated in the MaxNumberOfSamples field in the *StartSampling*
6950 command. If more samples are requested than can be delivered, the *GetSampleDataResponse*
6951 command will return the number of samples equal to MaxNumberOfSamples field. If fewer
6952 samples are available for the time period, only those available shall be returned.

6953 **Samples (mandatory):** Series of data samples captured using the interval specified by the
6954 *SampleRequestInterval* field in the *StartSampling* command. Each sample contains the change in
6955 the relevant data since the previous sample. Data is organised in a chronological order, the oldest
6956 sample is transmitted first and the most recent sample is transmitted last. Invalid samples should
6957 be marked as 0xFFFF.

6958 D.3.2.3.1.8.3 When Generated

6959 A *GetSampledDataResponse* command is generated in response to *GetSampledData* command.
6960 A ZCL Default Response with status NOT_FOUND shall be returned if there is no appropriate
6961 Sample data available.

6962

6963

6964 D.3.2.3.1.9 ConfigureMirror Command

6965 Where ‘Two Way Mirroring’ is being implemented, this command shall be sent to the mirror
6966 once the mirror has been created. The command allows a BOMD to provide the operational
6967 configuration of the associated Mirror. Note that this command is not required for a traditional
6968 ‘One way’ mirror (see D.3.4.4.3 for further details).

6969 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)

6970

Figure D-28– Format of the *ConfigureMirror* Command Payload

6971 D.3.2.3.1.9.2 Payload Details

6972 **Issuer Event ID (mandatory):** Unique identifier generated by the device being mirrored. When
6973 new information is provided that replaces older information, this field allows devices to

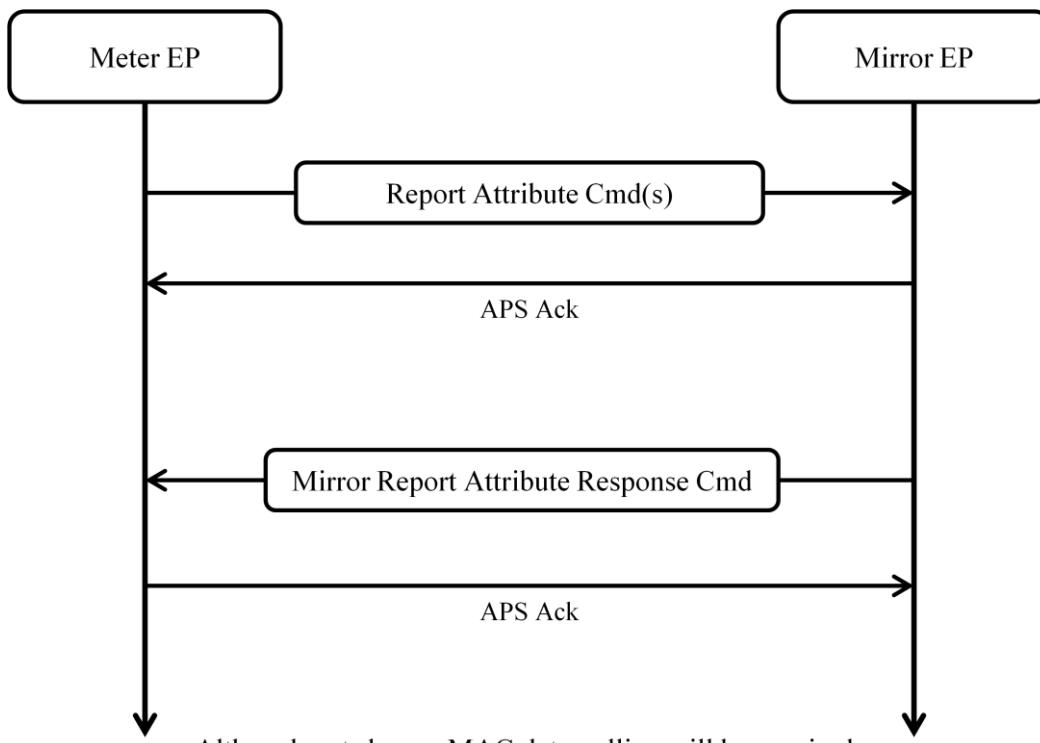
6974 determine which information is newer. It is recommended that the value contained in this field is
 6975 a UTC based time stamp (UTCTime data type) identifying when the command was issued. Thus,
 6976 newer information will have a value in the Issuer Event ID field that is larger than older
 6977 information.

6978 **Reporting Interval (mandatory):** An unsigned 24-bit integer to denote the interval, in seconds,
 6979 at which a mirrored meter intends to use the *ReportAttribute* command..

6980 **Mirror Notification Reporting (mandatory):** A Boolean used to advise a BOMD how the
 6981 Notification flags should be acquired (see below).

6982 When *Mirror Notification Reporting* is set, the *MirrorReportAttributeResponse* command is
 6983 enabled. In that case, the Metering client on the mirror endpoint shall respond to the last or only
 6984 *ReportAttribute* command with the *MirrorReportAttributeResponse*. This is shown if Figure
 6985 D-29:

6986



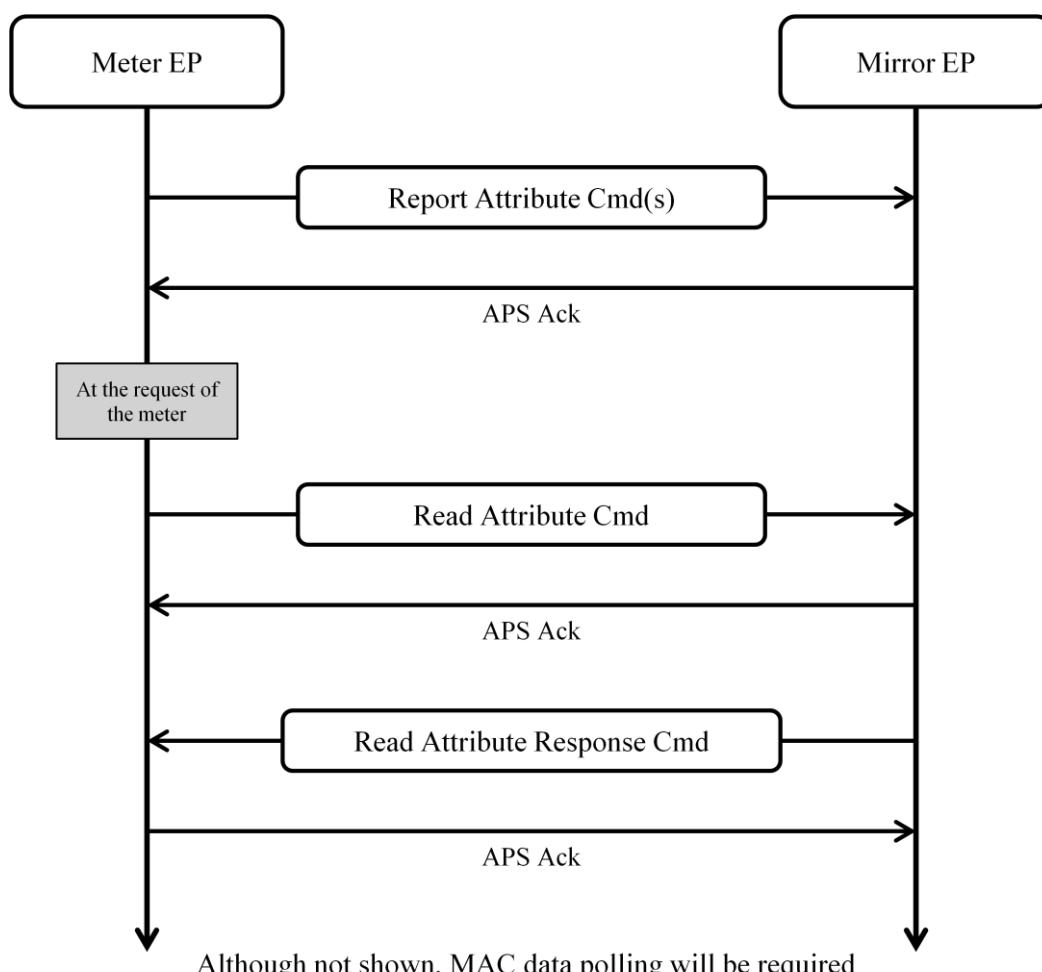
6987
 6988 **Figure D-29—*MirrorReportAttributeResponse* Command Enabled**

6989 **NOTES:**

- 6990 1. On powering up, the BOMD will send one or more *Report Attribute* commands to the
6991 Metering client on the mirror endpoint. The last attribute to be reported to the mirror shall be
6992 an *Attribute Reporting Status* attribute, as defined in section A.2.
- 6993 2. If *MirrorReportAttributeResponse* is enabled, the server does not need to request an APS
6994 ACK. If the server requests an APS ACK, the Metering client on the mirror endpoint shall
6995 respond first with an APS ACK and then send the *MirrorReportAttributeResponse*.

6996 If *Mirror Notification Reporting* is set to FALSE, the *MirrorReportAttributeResponse* command
6997 shall not be enabled; the Metering server may poll the Notification flags by means of a normal
6998 *ReadAttribute* command, as shown in Figure D-30:

6999



7000

7001

7002

Figure D-30—*MirrorReportAttributeResponse* Command Disabled

7003 **Notification Scheme (mandatory):** This unsigned 8-bit integer allows for the pre-loading of the
 7004 Notification Flags bit mapping to ZCL or Smart Energy Standard commands. The following
 7005 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

7006 **Figure D-31 – *NotificationScheme* Enumerations**

7007

7008 **D.3.2.3.1.9.3 When Generated**

7009 The *ConfigureMirror* command is generated in response to the *RequestMirrorResponse*
 7010 command when the Mirror has been created.

7011 **D.3.2.3.1.9.4 Effect on Receipt**

7012 On receipt of the *ConfigureMirror* command, the mirror will understand if the
 7013 *MirrorReportAttributeResponse* command should be sent, and if there is a scheme for the
 7014 Notifications flags. The Mirror will also understand the interval at which the Meter shall report to
 7015 the mirror.

7016 A ZCL Default Response with status **INVALID_FIELD** shall be returned if the required
 7017 **Notification Scheme** is not supported by the Mirroring device.

7018

7019 **D.3.2.3.1.10 *ConfigureNotificationScheme* Command**

7020 *Note:* The *ConfigureNotificationScheme* command in this revision of this specification is
 7021 provisional and not certifiable. This feature may change before reaching certifiable status in a
 7022 future revision of this specification.

7023 Where ‘Two Way Mirroring’ is being implemented, and a non-default Notification Scheme is to
 7024 be used, this command shall be sent to the mirror once the mirror has been created. The
 7025 command allows a BOMD to provide details of the required Notification Scheme to the
 7026 associated mirror, and should be used in conjunction with the associated
 7027 *ConfigureNotificationFlags* command (see D.3.2.3.1.11). No default schemes are allowed to be
 7028 overwritten (see D.3.4.4.3.4 and D.3.4.4.3.5 for further details); generic schemes should use one
 7029 of the reserved values 0x03 – 0x80, MSP schemes should use one of the values 0x081-0x0FE
 7030 (see Figure D-31). Section D.3.4.4.3.3 provides further details.

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

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.

7076 **Notification Scheme (mandatory):** An unsigned 8-bit integer that allows for the pre-loading of
7077 the Notification Flags bit mapping to ZCL or Smart Energy Standard commands. Figure D-31
7078 details the schemes that are currently supported within the Smart Energy Standard.

7079 **Notification Flag Attribute ID (mandatory):** An unsigned 16-bit integer that denotes the
7080 attribute id of the Notification flag (2-8) that will be configured for this Notification scheme.

7081 **Bit Field Allocation (mandatory):** The bit field allocation sub payload is defined in Figure
7082 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

7083 **Figure D-34– Bit Field Allocation Command Sub Payload**

7084 **Cluster ID (mandatory):** An unsigned 16-bit integer that denotes the Cluster id of the
7085 Notification flag that will be configured for this Notification scheme.

7086 **Manufacturer Code (mandatory):** An unsigned 16-bit integer that denotes the Manufacturer
7087 Code to be used with these command IDs, that are configured for this Notification flag within
7088 this Notification scheme.

7089 **No of Commands (mandatory):** An unsigned 8-bit integer that indicates the number of
7090 command identifiers contained within this sub payload.

7091 **Command ID (mandatory):** An unsigned 8-bit integer that denotes the command that is to be
7092 used. The command id should be used with the cluster id to reference the command(s).

7093 *D.3.2.3.1.11.3 When Generated*

7094 This command is sent once the mirror has been created, and the *ConfigureNotificationScheme*
7095 command has been sent up the top level of the scheme. There is a *ConfigureNotificationFlags*
7096 command for each attribute that the scheme is proposing to use. No default schemes are allowed
7097 to be overwritten.

7098 *D.3.2.3.1.11.4 Effect on Receipt*

7099 Once all *ConfigureNotificationFlags* commands have been received, a fully populated scheme
7100 will be available, and two way mirroring can then be enabled.

7101

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.

- 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 commodity provider to whom this command relates.

7139 **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.

7145 **Implementation Date/Time (mandatory):** A UTC Time field to indicate the date at which the originating command was to be applied.

7147 **Supply Status (mandatory):** An 8-bit enumeration field indicating the status of the energy supply controlled by the Metering Device following implementation of the originating command. 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

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

7178 most significant Octet specifies the attribute set and the least significant Octet specifies the
7179 attribute within the set. The currently defined attribute sets are listed in Table D-57.

7180 **Table D-57 – Metering Cluster Client Attribute Sets**

Attribute Set Identifier	Description
0x00	Notification Attribute Set
0x01 to 0xFF	Reserved

7181

7182 **D.3.3.2.1 Notification Attribute Set**

7183 The Notification Attribute Set is used to notify battery operated mirrored devices (BOMDs) that
7184 the ESI or other HAN device has pending information which should be fetched.

7185 Only clients on a mirror endpoint shall support this attribute set.

7186 When commands / attributes are received into the ESI from the HES or other HAN devices, the
7187 ESI will store the corresponding information and set the appropriate bits in the Notification Flag
7188 attributes (BitMaps). The ESI shall reset the bit as soon as a ‘Get’ command with the
7189 corresponding message type is received and all commands of the appropriate type have been
7190 retrieved (this is to allow for multiple commands of the same type).

7191 **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

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

7200 **New OTA Firmware Flag:** will be set by the ESI, when the ESI has new OTA Firmware to send
7201 to the BOMD. The BOMD can then make the decision on when it starts the OTA upgrade
7202 request.

7203 **CBKE Update Request Flag:** requests the BOMD to initiate the CBKE process with the Trust
7204 Center to replace the link key currently in use.

7205 **Time Sync Request Flag:** requests the BOMD to initiate the time synchronization process with
7206 the Time server.

7207 **The Stay Awake Request Flags:** will be set by the ESI when the ESI wants to send a command.
7208 There are two types of the Stay Awake requests, one for HAN requests and one for WAN
7209 requests; an implementation may react differently depending on the source of the request:

7210 **HAN requests:** The HAN Stay Awake flag should only be used for commands that
7211 originate from HAN and not from the ESI that is supporting the commodity of the
7212 Mirrored device.

7213 **WAN requests:** The WAN Stay Awake flag should only be used for commands that
7214 originate from the backhaul network.

7215 **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

7216
7217 Table D-60 shows example usage of the Stay Awake Request flags. The most likely use is
7218 when profile data or snapshots are required. The commands shown require additional
7219 parameters to be sent by the requesting device to solicit the correct response from the
7220 receiver.

7221 A battery operated meter should read the Notification Flags regularly or enable the
7222 *MirrorReportAttributeResponse* command. If the *StayAwakeRequest* flag is set, the
7223 battery operated meter shall poll its parent node at least three times for pending messages.
7224 The polling interval shall be configurable and not less than 250 ms.

7225 The ESI shall try to send commands after the BOMD has pushed meter readings or read
 7226 the Notification Flags. After reception of a command, the BOMD shall read the
7227 *NotificationFlags* again. If the ESI has successfully transmitted all pending commands, it
 7228 shall reset the StayAwakeRequest flag.

7229 Nevertheless, the BOMD can decide to go to sleep if the *StayAwakeRequest* flag is not
7230 reset after consecutive reads of the *NotificationFlags* attribute or if it is required by its
 7231 power supply constraints.

7232 **Push Historical Metering Data Attribute Set:** This notification flag requests the BOMD to
7233 push a sub set of the historical consumption information found within the Metering cluster's
 7234 'Historical Consumption Attribute Set'. The format of the bits is defined within Table D-61.

7235 **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

7236 **7237 Push Historical Payment Data attribute Set:** This notification flag requests the BOMD to push
7238 a sub set of the historical consumption cost information found within the Prepayment cluster's
 7239 'Historical Cost Consumption Attribute Set'. The format of the bits is defined within Table D-62.

7240 **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

7241 **7242 Push All Static Data - Basic Cluster:** This notification flag requests the BOMD to push all of
 7243 the attributes within the Basic cluster that are supported by the mirrored meter.

- 7244 **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.
- 7245
- 7246 **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.
- 7247
- 7248 **Network Key Active:** When this notification flag has been set, **the meter shall check with the TC to update the network key.**
- 7249
- 7250 **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).
- 7251
- 7252 **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).
- 7253
- 7254
- 7255 **Change Supply Message:** When this notification flag has been set, **the meter shall send a *GetNotifiedMessage* command to the mirror.**
- 7256
- 7257 **Local Change Supply Message:** When this notification flag has been set, **the meter shall send a *GetNotifiedMessage* command to the mirror.**
- 7258
- 7259 **SetUncontrolledFlowThreshold Message:** When this notification flag has been set, **the meter shall send a *GetNotifiedMessage* command to the mirror.**
- 7260
- 7261 **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.**
- 7262
- 7263
- 7264 **Get Snapshot Message:** When this notification flag has been set, **the meter shall send a *GetNotifiedMessage* command to the mirror.**
- 7265
- 7266 **Get Sampled Data Message:** When this notification flag has been set, **the meter shall send a *GetNotifiedMessage* command to the mirror.**
- 7267
- 7268
- 7269 D.3.3.2.1.2 NotificationFlags Attributes
- 7270 NotificationFlags2 to NotificationFlags8 are 32-bit bitmaps that each represent a series of flags.
- 7271 **Each flag represents an outstanding command that the Mirror is holding on behalf of the BOMD.**
- 7272 **Each flag represents a different command.** The **format of these attributes is dictated by the scheme that is currently in operation.**
- 7273
- 7274

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	<i>Get Profile</i>	O
0x01	<i>Request Mirror Response</i>	O
0x02	<i>Mirror Removed</i>	O
0x03	<i>Request Fast Poll Mode</i>	O
0x04	<i>ScheduleSnapshot</i>	O
0x05	<i>TakeSnapshot</i>	O
0x06	<i>GetSnapshot</i>	O
0x07	<i>StartSampling</i>	O
0x08	<i>GetSampledData</i>	O
0x09	<i>MirrorReport AttributeResponse</i>	O
0x0A	<i>ResetLoadLimit Counter</i>	O
0x0B	<i>Change Supply</i>	O
0x0C	<i>Local Change Supply</i>	O
0x0D	<i>SetSupplyStatus</i>	O
0x0E	<i>SetUncontrolledFlowThreshold</i>	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

Octets	1	4	1
Data Type	8-bit Enumeration ^a	UTCTime	Unsigned 8-bit Integer
Field Name	Interval Channel	End Time	NumberOfPeriods

7284 a. CCB 1077

7285 **Figure D-38– Get Profile Command Payload**

7286 **D.3.3.3.1.1.1 Payload Details**

7287 **Interval Channel:** Enumerated value used to select the quantity of interest returned by the
7288 *GetProfileReponse* command. The Interval Channel values are listed in Table D-64.

7289 **Table D-64– Interval Channel Values**

Enumerated Value	Description
0	Consumption Delivered
1	Consumption Received
2 to 255	Not used

7290
7291 **EndTime:** 32-bit value (in UTCTime) used to select an Intervals block from all the Intervals
7292 blocks available. The Intervals block returned is the most recent block with its EndTime
7293 equal or older to the one provided. **The most recent Intervals block is requested using an**
7294 **End Time set to 0x00000000**, subsequent Intervals block are requested using an End time set to
7295 the EndTime of the previous block - (number of intervals of the previous block *
7296 ProfileIntervalPeriod).

7297 **NumberofPeriods:** **Represents the number of intervals being requested.** This value cannot
7298 exceed the size stipulated in the *MaxNumberOfPeriodsDelivered* attribute. If more intervals are
7299 requested than can be delivered, the *GetProfileResponse* will return the number of intervals equal
7300 to *MaxNumberOfPeriodsDelivered*. If fewer intervals are available for the time period, only
7301 those available are returned.

7302 **D.3.3.3.1.1.2 When Generated**

7303 **The *GetProfile* command is generated when a client device wishes to retrieve a list of captured**
7304 **Energy, Gas or water consumption for profiling purposes.** Due to the potentially large amount
7305 of profile data available, **the client device should store previously gathered data and only**
7306 **request the most current data.** When initially gathering significant amounts of historical
7307 interval data, the *GetProfile* command should not be issued any more frequently than 7.5
7308 seconds to prevent overwhelming the ZigBee network.

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:

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

Figure D-40—*Mirror Removed* Command Payload

7337

D.3.3.3.1.3.2 Payload Details

7338 **Removed EndPoint ID:** 16 Bit Unsigned Integer indicating the End Point ID previously
7339 containing the Metering Devices meter data.

7340

7341 D.3.3.3.1.4 Request Fast Poll Mode Command

7342 D.3.3.3.1.4.1 Payload Format

7343 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

7345

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.

7348 Duration: Desired duration for the server to remain in fast poll mode not to exceed 15
7349 minutes as specified in sub-clause D.3.4.2.

7350 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).

7355 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*.

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.

7384 **Snapshot Start Time (mandatory):** The *Snapshot Start Time* denotes the date/time when
7385 the Snapshot schedule is to start.

7386 **Snapshot Schedule (mandatory):** A 24-bit value indicating the schedule that should be
7387 used for the snapshot. The snapshot schedule bit field is formatted as indicated in Table
7388 D-65.

7389 **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

7390 **SnapshotPayloadType (mandatory):** The *SnapshotPayloadType* is an 8-bit enumerator
7391 defining the format of the *SnapshotPayload* required. The different snapshot types are
7392 listed in Table D-53. The server selects the *SnapshotPayloadType* based on the charging
7393 scheme in use.

7394 **Snapshot Cause (mandatory):** A 32-bit BitMap indicating the cause of the snapshot.
7395 The snapshot cause values are listed in Table D-52.

7396

7397 **D.3.3.3.1.6 TakeSnapshot Command**

7398 This command is used to instruct the cluster server to take a single snapshot. See section D.3.4.5
7399 for further details.

7400 **D.3.3.3.1.6.1 Payload Format**

Octets	4
Data Type	32-bit BitMap
Field Name	Snapshot Cause (M)

7401 **Figure D-44—TakeSnapshot Command Payload**

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.

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 case¹⁵.

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

7458 **SampleType (mandatory):** An 8 bit enumeration that identifies the type of data being sampled.
 7459 Possible values are defined in Table D-54.

7460 **SampleRequestInterval (mandatory):** An unsigned 16-bit field representing the interval or
 7461 time in seconds between samples.

7462 **MaxNumberofSamples (mandatory):** A 16 bit unsigned integer that represents the number of
 7463 samples to be taken.

7464 **D.3.3.3.1.8.3 Effect on Receipt**

7465 On receipt of the *StartSampling* command, the Metering Device shall respond with a
 7466 *StartSamplingResponse* command indicating the *Sample ID* allocated to this Sampling session. If
 7467 the Metering Device is unable to support a further Sampling session, *Sample ID* shall be
 7468 returned as 0xFFFF. See D.3.2.3.1.14 for further details.

7469

7470 **D.3.3.3.1.9 GetSampledData Command**

7471 This command is used to request sampled data from the server. Note that it is the responsibility
 7472 of the client to ensure that it does not request more samples than can be held in a single command
 7473 payload.

7474 **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)

7475 **Figure D-47– GetSampledData Command Payload**

7476 **D.3.3.3.1.9.2 Payload Details**

7477 **SampleID (mandatory):** Unique identifier allocated to this Sampling session. This field allows
 7478 devices to match response data with the appropriate request. See D.3.2.3.1.14 for further details.

7479 **EarliestSampleTime (mandatory):** A UTC Timestamp indicating the earliest time of a sample
 7480 to be returned. Samples with a timestamp equal to or greater than the specified
 7481 *EarliestSampleTime* shall be returned.

7482 **SampleType (mandatory):** An 8 bit enumeration that identifies the required type of sampled
 7483 data. Possible values are defined in Table D-54.

7484 **NumberOfSamples (mandatory):** Represents the number of samples being requested. This
 7485 value cannot exceed the size stipulated in the *MaxNumberOfSamples* field in the *StartSampling*
 7486 command. If more samples are requested than can be delivered, the *GetSampledDataResponse*
 7487 command will return the number of samples equal to the *MaxNumberOfSamples* field. If fewer
 7488 samples are available for the time period, only those available are returned.

7489 **D.3.3.3.1.9.3 Effect on Receipt**

7490 On receipt of this command, the server shall respond with a *GetSampledDataResponse* command
 7491 containing the samples meeting the selection criteria detailed in this command. Details of the
 7492 *GetSampledDataResponse* command are detailed in sub-clause D.3.2.3.1.8.

7493 A ZCL Default Response with status NOT_FOUND shall be returned if the server does not have
 7494 sample data which satisfies the received parameters.

7495

7496 **D.3.3.3.1.10 MirrorReportAttributeResponse Command**

7497 This command is sent in response to the *ReportAttribute* command when the *MirrorReporting*,
 7498 attribute is set.

7499 **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)

7500 **Figure D-48– MirrorReportAttributeResponse Command Payload**

7501 **D.3.3.3.1.10.2 Payload Details**

7502 The payload of this command is defined within the *ConfigureNotificationScheme* command.

7503 **Notification Scheme (mandatory):** An unsigned 8-bit integer that allows for the pre-loading of
 7504 the Notification Flags bit mapping to ZCL or Smart Energy Standard commands. Figure D-31
 7505 details the schemes that are currently supported within the Smart Energy Standard.

7506 **Notification Flags #N (mandatory):** see sections D.3.3.2.1.1 and D.3.3.2.1.2.

7507 **D.3.3.3.1.10.3 When Generated**

7508 The *MirrorReportAttributeResponse* command is generated in response to the *ReportAttribute*
 7509 command when the *MirrorReporting* attribute is set. The *MirrorReportAttributeResponse*
 7510 command is sent from the Mirror to the meter.

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

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

7535

Figure D-50– Change Supply Command Payload

7536 **D.3.3.3.1.12.2 Payload Details**

7537 **Provider ID (mandatory):** An unsigned 32-bit field containing a unique identifier for the
7538 commodity provider to whom this command relates.

7539 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. When
7540 new information is provided that replaces older information for the same time period, this field
7541 allows devices to determine which information is newer. The value contained in this field is a
7542 unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)
7543 identifying when the command was issued. Thus, newer information will have a value in the
7544 *Issuer Event ID* field that is larger than older information.

7545 **Request Date/Time (mandatory):** A UTC Time field to indicate the date and time at which the
7546 supply change was requested.

7547 **Implementation Date/Time (mandatory):** A UTC Time field to indicate the date at which the
7548 supply change is to be applied. An *Implementation Date/Time* of **0x00000000** shall indicate that
7549 the command should be executed immediately. An *Implementation Date/Time* of **0xFFFFFFFF**
7550 shall cause an existing but pending *Change Supply* command with the same *Provider ID* and
7551 *Issuer Event ID* to be cancelled (the status of the supply will not change but the *Proposed*
7552 *Change Supply Implementation Time* attribute shall be reset to zero).

7553 **Proposed Supply Status (after Implementation):** An 8-bit enumeration field indicating the
7554 status of the energy supply controlled by the Metering Device following implementation of this
7555 command. The enumerated values for this field are outlined in Table D-56.

7556 **Supply Control Bits:** An 8-bit BitMap where the least significant nibble defines the Supply
7557 Control bits, the encoding of which is outlined in Table D-66:

7558

Table D-66– Supply Control Bits

Bits	Description
0	Acknowledge Required
1	Reserved
2	Reserved
3	Reserved

7559

7560 **Acknowledge Required:** Indicates that a *Supply Status Response* command is to be sent in
7561 response to this command. Note that the *Supply Status Response* command will only be returned
7562 to the originator when the *Change Supply* command has been successfully executed.

7563 **D.3.3.3.1.12.3 When Generated**

7564 A Head-end or ESI may send an INTERRUPT, ARM or (if allowed) RESTORE command to a
7565 metering device.

7566 The execution of an INTERRUPT or ARM command may be delayed, as indicated by the
7567 Implementation Date/Time field; these commands shall only come from a Head-End via an ESI.
7568 A subsequent command with a new Implementation Date/Time shall override an existing delayed
7569 command. A new command with an Implementation Date/Time of 0x00000000 shall be executed
7570 immediately, but shall not cancel an existing delayed command; to override an existing delayed
7571 command with a command to be executed immediately, a command to cancel the existing
7572 command should first be sent followed by the new command to be executed immediately (see
7573 notes on *Implementation Date/ Time* field in D.3.3.3.1.12.2 for further details).

7574 The addition of credit or selection of Emergency credit shall not cause a delayed INTERRUPT
7575 command to be cancelled (these will be cancelled by the Head-End and a new supply control
7576 command sent down).

7577 **D.3.3.3.1.12.4 Effect on Receipt**

7578 If required, a *Supply Status Response* command shall be returned to the originator when the
7579 *Change Supply* command has been successfully executed (see D.3.2.3.1.13 for further details).

7580 A ZCL response, indicating ‘Unauthorized’ (NOT_AUTHORIZED), shall be immediately
7581 returned to an originator requesting a supply change that is not allowed in the current application.

7582 A ZCL response, indicating ‘Unavailable’ (UNSUP_CLUSTER_COMMAND), shall be
7583 immediately returned to an originator requesting a supply change by a metering device that is
7584 incapable of carrying out the action (e.g. an INTERRUPT command to a metering device that has
7585 no contactor).

7586 A ZCL response, indicating INVALID_VALUE, shall be immediately returned to an originator
7587 requesting a supply change containing a non-zero Implementation Date/Time that is less than or
7588 equal to the current date/time (i.e. is in the past).

7589

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

7614 D.3.3.3.1.14 SetSupplyStatus Command

7615 This command is used to specify the required status of the supply following the occurrence 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

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.

7656 **Uncontrolled Flow Threshold (mandatory):** The threshold above which a flow meter (e.g. Gas
 7657 or Water) shall detect an uncontrolled flow. A value of 0x0000 indicates the feature in unused.

7658 **Unit of Measure (mandatory):** An enumeration indicating the unit of measure to be used in
 7659 conjunction with the *Uncontrolled Flow Threshold* attribute. The enumeration used for this field
 7660 shall match one of the UnitOfMeasure values using a pure binary format as defined in the
 7661 Metering cluster (see sub-clause D.3.2.2.4.1).

7662 **Multiplier (mandatory):** An unsigned 16-bit value indicating the multiplier, to be used in
 7663 conjunction with the *Uncontrolled Flow Threshold* and *Divisor* fields, to determine the true flow
 7664 threshold value. A value of 0x0000 is not allowed.

7665 **Divisor (mandatory):** An unsigned 16-bit value indicating the divisor, to be used in conjunction
 7666 with the *Uncontrolled Flow Threshold* and *Multiplier* fields, to determine the true flow threshold
 7667 value. A value of 0x0000 is not allowed.

7668 **Stabilisation Period (mandatory):** An unsigned 8-bit value indicating the time given to allow
 7669 the flow to stabilize. It is defined in units of tenths of a second.

7670 **Measurement Period (mandatory):** An unsigned 16-bit value indicating the period over which
 7671 the flow is measured and compared against the *Uncontrolled Flow Threshold* value. It is defined
 7672 in units of 1 second.

7673
 7674
 7675

7676 D.3.4 Metering Application Guidelines

7677 D.3.4.1 Attribute Reporting

7678 Attribute reporting may be used for sending information in the Reading Information,
 7679 TOU Information, Meter Status, and Historical Consumption attribute sets. Use of the
 7680 Report Attribute command without report configuration may be used for unsolicited notification
 7681 of an attribute value change. Sleepy devices may have to poll.

7682 D.3.4.2 Fast Polling or Reporting for Monitoring Energy Savings

7683 Client devices, such as an energy gateway, smart thermostat, or in-home displays can monitor
 7684 changes to energy saving settings within the premises and give users near real time feedback and
 7685 results. The Metering cluster can support this by using Attribute Reporting and sending updates
 7686 at a much faster rate for a short period of time. Client devices can also perform a series of
 7687 Attribute reads to accomplish the same task. In either case, requests or updates shall be limited
 7688 to a maximum rate of once every two seconds for a maximum period of 15 minutes. These

7689 limitations are required to ensure Smart Energy profile based devices do not waste available
7690 bandwidth or prevent other operations within the premises.

7691 **D.3.4.3 Metering Data Updates**

7692 The frequency and timeliness of updating metering data contained in the Metering
7693 Cluster attributes and Profile Intervals is up to the individual Metering device manufacturer's
7694 capabilities. As a best practice recommendation, updates of the metering data should not cause
7695 delivery of the information to end devices more often than once every 30 seconds. End
7696 devices should also not request information more often than once every 30 seconds. The Fast
7697 Polling attributes and commands shall be used by client devices requesting information more
7698 often than once every 30 seconds.

7699 **D.3.4.3.1 Fast Polling Periods**

7700 Since the *DefaultUpdatePeriod* specifies the normal update interval and *FastPollUpdatePeriod*
7701 specifies the fastest possible update interval, it is recommended that metering clients read these
7702 attributes to determine the optimal normal/fast polling interval and the optimal fast poll period
7703 to request. Client devices shall not request data more frequent than *FastPollUpdatePeriod* or
7704 the *AppliedUpdatePeriod*.

7705 **D.3.4.4 Mirroring**

7706 The SE Profile specifies Mirror support in the Metering cluster to store and provide access to
7707 data from metering devices on battery power. Devices with resources to support mirroring
7708 advertise the capability using the *Basic Attribute Physical Environment*.

7709 **D.3.4.4.1 Discovery**

7710 The SE standard does not prescribe how Mirroring is implemented. Devices may query the
7711 Basic Cluster attribute *PhysicalEnvironment* to determine Mirrored device capacity prior to
7712 CBKE (see sub-clause D.3.4.4.2 below). This would allow a battery based end device to
7713 discover if an ESI has capacity to mirror data prior to the process of joining the network in a
7714 secure manner, thereby reducing retry attempts. This would also enhance the service discovery
7715 of the ZDO Match Descriptor that would be used to determine if an endpoint can request the
7716 setup and removal of a mirrored Metering cluster. Once a device has joined the network and
7717 performed CBKE, it can then request setup of a mirrored metering cluster. ZDO Discovery
7718 should be supported to allow HAN devices to discover the mirror endpoints; only active mirror
7719 endpoints shall be discoverable. This process may need to be repeated in the case of a Trust
7720 Center swap-out (refer to sub-clause 5.4.2.2.3 for further information).

7721 **D.3.4.4.2 Mirror Attributes**

7722 The mandatory *Basic*, *Metering*, and (where applicable) *Prepayment* attributes shall be
7723 supported. The Basic Cluster *PhysicalEnvironment* attribute shall be supported on ESIs
7724 supporting mirroring functionality; an enumerated value of 0x01 would indicate that the
7725 device has the capacity to mirror an end device; a value of 0x00 would specify an "Unspecified

7726 environment" per the ZCL specification. Only the Basic cluster for devices capable of
7727 providing a mirror shall have the *PhysicalEnvironment* attribute set to 0x01. The **ZCL Report**
7728 **Attribute** command shall be used to push data to the mirror. Only the metering device that has
7729 been granted a mirror on a certain endpoint is allowed to push data to that endpoint. The ZCL Not
7730 Authorized return status shall be used to provide access control. The use of ZCL Report
7731 Configuration shall not be required to generate *Report Attribute Command*.

7732 Manufacturers will design and manufacture devices to meet customer requirement specifications
7733 that will state the functionality of the battery powered meter and therefore devices supporting
7734 mirroring in the field will also have to support those requirements through an appropriate choice
7735 of optional attributes. Battery powered devices will report attributes to the mirror as required by
7736 the customer specification. In the event that the mirror is out of memory space or cannot
7737 support the attribute it shall respond **ATTRIBUTE_UNSUPPORTED** back to the battery-
7738 powered meter. The same response (**ATTRIBUTE_UNSUPPORTED**) will be sent to a device
7739 querying the mirror for an attribute it doesn't support. A device querying the mirror for an
7740 attribute that is supported but not yet available (the battery powered meter hasn't yet sent the
7741 attribute) shall receive a response **ATTRIBUTE_UNAVAILABLE** from the mirror.

7742 **D.3.4.4.3 Two Way Mirror for BOMD**

7743 The primary purpose of a mirror is to present data from a sleepy battery operated mirrored device
7744 (**BOMD**), to a HAN, when communication to the **BOMD** is not available. However, there is also
7745 a need to pass data to the **BOMD** in these circumstances.

7746 In ZigBee terms, the device providing a mirror has to be a trusted device. There will be **APS**
7747 security between originating device and mirror, and between mirror and **BOMD**. Messages that
7748 require end-to-end security must be secured by other means.

7749 Any device on the HAN wishing to communicate with a **BOMD** must do so via the mirror. The
7750 mirror and the **BOMD** must support the Notification Attribute Set of the Metering cluster,
7751 designed to allow the **BOMD** to establish if there any messages waiting on the mirror for
7752 collection. There are 4 mechanisms provided to allow information destined for a **BOMD** to be
7753 transferred to the **BOMD**:

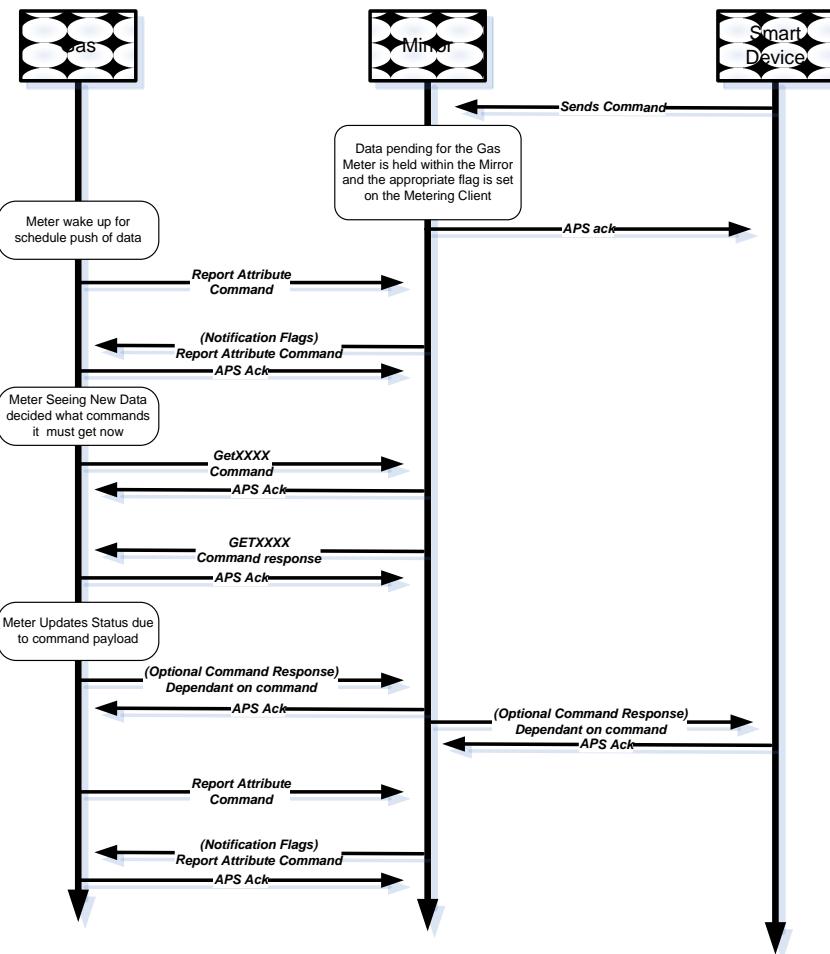
- 7754 1 For several required actions, the Notification Flag conveys all required information. Many of
7755 the bits within the *Functional Notification Flags* attribute utilize this method; the **Push All**
7756 **Static Data - Metering Cluster** bit is an example of this.
- 7757 2 For those clusters where the **BOMD** is a client (e.g. Price, Calendar, Device Management),
7758 the flags in the Notification Attribute Set allow the **BOMD** to quickly determine if there is
7759 any new information of interest to the **BOMD**. This information is normally sent from the
7760 backhaul (i.e. the Head End System). Upon waking, and having acquired the status of the
7761 relevant Notification Flags, the **BOMD** will fetch those commands matching the set flags by

7762 sending the appropriate ‘get’ command(s) **to the associated ESI endpoint**. (Note that the
7763 associated data will usually be held on the device providing the ESI; therefore the associated
7764 get/publish commands could be utilized multiple times). It is **recommended that a ‘binding-**
7765 **type’ mechanism is used internally within the mirroring device to link the ESI and Mirror**
7766 **endpoints.**

- 7767 3 Where a cluster server is located on the BOMD (e.g. Metering, Prepayment and Basic
7768 clusters), pre-specified transient commands sent from cluster clients will have to be buffered
7769 on the Mirror until such time as the BOMD awakes and can fetch them. Upon waking, and
7770 having acquired the status of the relevant Notification Flags, the BOMD will fetch those
7771 commands matching the set flags by sending a *GetNotifiedMessage* command **to the Mirror**
7772 **endpoint**. In this case, the Mirror shall remember the address of the device initially
7773 originating the command, so that any response can be returned, via the mirror, to that device.
7774 As an example, the handling of a (Prepayment cluster) *Consumer Top Up* command utilizes
7775 this method.
- 7776 4 Non-specified transient commands, destined for cluster servers located on a BOMD, will
7777 also have to be buffered on the Mirror until such time as the BOMD awakes. In this case, the
7778 appropriate ‘Stay Awake’ Notification flag will be set (to advise the BOMD to remain
7779 awake for a longer period) and, once the mirroring device recognizes that the BOMD is
7780 awake, it shall attempt to push those buffered commands to the BOMD as soon as possible.
7781 Attempts to transfer these commands shall be repeated until such time as the command(s)
7782 is/are successfully moved to the BOMD (this may not be within the same BOMD wake
7783 period). The Mirror shall again remember the address of the device initially originating the
7784 command, so that any response can be returned, via the mirror, to that device. As an
7785 example, the handling of a (Metering cluster) *Get Profile* command utilizes this method.

7786 .

7787



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

7798 2 The mirror “caches” the command and sets the appropriate notification flag, to signal that
7799 data is waiting. The mirror also returns a ZCL Default Response to the initiating device with
7800 a status code of NOTIFICATION_PENDING. If the command buffer on the mirror is
7801 already full, the mirror shall instead return a ZCL Default Response to the initiating device
7802 with a status code of INSUFFICIENT_SPACE.

7803 3 The gas meter wakes up and polls for a notification

7804 4 The notification is returned

7805 5 This may be all that is required (e.g. a request to update static data on the mirror).

7806 6 The meter requests the data according to the Notification flag that was raised

7807 7 The mirror sends the command that was originally received from the IHD

7808 8 The meter may update data on the mirror in order to indicate to the device initiating the
7809 command that its action has been carried out.

7810 A mirror that caches a command on behalf of a HAN device, prior to that command being sent to
7811 a BOMD, may choose to time out that command after an appropriate period of time. The timeout
7812 period may be configurable based on the operator of the network and is not defined by this
7813 specification. If the mirror chooses to timeout a cached command, then it shall send a ZCL
7814 Default Response to the originator of the message with the same Transaction Sequence Number
7815 as received in the cached command. The Default Response shall contain the status code of
7816 TIMEOUT.

7817

7818 D.3.4.4.3.1 Responses to an Initiating Device

7819 Commands that have been buffered on a Mirror may trigger ZCL Default or command-specific
7820 responses to be returned once the command has reached and been actioned by the BOMD. In
7821 turn, these responses should be communicated to the device originally initiating the buffered
7822 command.

7823 In order to ensure that these responses are correctly relayed back to the device initiating the
7824 original command, it is recommended that:-

7825 1 The mirroring device, if supporting Two-way Mirroring, be able to store information that
7826 can be used to track the originator of a command stored in the buffer when the command is
7827 retrieved by the BOMD. This information should include, as a minimum, the device address,
7828 endpoint and Transaction Sequence Number (TSN) of the original command, for the
7829 purposes of relaying a response back to the originator. The TSN used when the buffered
7830 command is forwarded to the BOMD should also be included in the stored information for
7831 the purpose of matching the information to any associated response. The BOMD shall
7832 include the TSN of a forwarded buffered command in any associated response. A TSN in the
7833 ZCL header is only 8 bits and this may not provide enough information to produce a unique

7834 ID (or unique enough); if this is considered to be insufficient information, the mirroring
 7835 device could also store the cluster and command id.

7836 2 The mirroring device should relay the ZCL payload of the buffered command to the BOMD
 7837 using a newly generated ZCL command (i.e. using the address and security associated with
 7838 the BOMD).

7839 3 The payload of a response from the BOMD to the retrieved buffered command, should be
 7840 relayed to the originating device in a newly generated ZCL response which utilizes the
 7841 information stored by the mirroring device as detailed in point 1. This shall include the TSN
 7842 of the command received from the originating HAN device.

7843 4 All commands retrieved by the BOMD using the notification flag mechanism should support
 7844 a ZCL default response in cases where an explicit response is not defined (this is the normal
 7845 mode of operation; however the default response can be disabled).

7847 D.3.4.4.3.2 Unsolicited Commands from a BOMD

7848 Where a command is sent unsolicited from a cluster server on a BOMD, the BOMD should
 7849 publish that command to the mirror and the mirror should then publish that command to all
 7850 associated client devices that have bound to the respective server on the mirror.

7851 Client devices wishing to receive unsolicited commands published from a BOMD shall bind to
 7852 the respective server(s) on the BOMD mirror.

7854 D.3.4.4.3.3 Configuring a Two Way Mirror

7855 1 When utilizing a two-way mirror with a BOMD, certain configuration data must be passed from the
 7856 BOMD to the mirror once the mirror endpoint has been activated:-Under normal circumstances,
 7857 a BOMD will utilize one of the predefined Notification Schemes that will be pre-loaded onto
 7858 the mirror. In this case, only a *ConfigureMirror* command will be required; this command
 7859 will advise the mirror of the reporting interval to be used, the required mechanism to be used
 7860 to acquire Notification Flag status, and the predefined Notification Scheme to be used.

7861 2 Predefined Notification schemes cannot be modified. If a BOMD wishes to modify an
 7862 existing predefined scheme, or utilize a new generic or MSP Notification Scheme, then the
 7863 associated two-way mirror must be configured with information defining the new scheme
 7864 before that scheme can be used. New generic schemes should use one of the reserved values
 7865 0x03–0x80, MSP schemes should use one of the values 0x081-0x0FE (see Figure D-31).

7866 A BOMD can configure a new Notification Scheme on a mirror once that mirror has been
 7867 created (endpoint known). The BOMD shall send a *ConfigureNotificationScheme* command

7868 to the mirror, together with associated *ConfigureNotificationFlags* command(s), before
7869 transmitting a *ConfigureMirror* command that utilizes the new Notification Scheme. On
7870 receipt of the *ConfigureNotificationScheme* command, the mirror shall store the
7871 NotificationScheme information, and wait for the associated *ConfigureNotificationFlags*
7872 commands. Until all of the *ConfigureNotificationFlags* commands have been received, the
7873 two-way mirror functionality should be disabled. The Notification Flag Order parameter in
7874 the *ConfigureNotificationScheme* command will allow the mirror to determine when all of
7875 the *ConfigureNotificationFlags* commands have been received.

7876

7877 D.3.4.4.3.4 Predefined Notification Scheme A

7878 Notification Scheme A is a predefined scheme for the order of the bit strings within each of the
7879 *NotificationFlag#N* attributes. See sections D.3.2.3.1.10 and D.3.2.3.1.11 for configuration of
7880 other schemes. Refer to section D.3.4.4.3 for details on the usage of these Notification Flags.

7881 **D.3.4.4.3.4.1 MirrorReportAttributeResponse Command Format**

7882 The format for Notification Scheme A is **0x0FFFFFFF** meaning that the first and only
7883 Notification flag to be transmitted within the *MirrorReportAttributeResponse* command will be
7884 the *FunctionalNotificationFlags* attribute.

7885 **FunctionalNotificationFlags Attribute:** Defined in section D.3.3.2.1.1.

7886

7887 D.3.4.4.3.5 Predefined Notification Scheme B

7888 Notification Scheme B is a predefined scheme for the order of the bit strings within each of the
7889 *NotificationFlag#N* attributes. See sections D.3.2.3.1.10 and D.3.2.3.1.11 for configuration of
7890 other schemes. Refer to section D.3.4.4.3 for details on the usage of these Notification Flags.

7891 **D.3.4.4.3.5.1 MirrorReportAttributeResponse Command Format**

7892 The format for Notification Scheme B is **0x01234FFF** meaning the first Notification flag to be
7893 transmitted within the *MirrorReportAttributeResponse* command will be the
7894 *FunctionalNotificationFlags* attribute followed by *NotificationFlags2* to *NotificationFlags5*.

7895 **FunctionalNotificationFlags Attribute:** Defined in section D.3.3.2.1.1.

7896 **NotificationFlags2 Attribute:** The *NotificationFlags2* attribute shall be configured to support
7897 the Price cluster and is implemented as a set of bit flags which are defined as shown below:

7898

7899 **Table D-69– Notification Flags 2**

Bit Number	Waiting Command
------------	-----------------

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

7900 ^a A *Publish Price* command may result from more than one ‘Get’ command; for clarity, a *GetCurrentPrice* command
 7901 should be sent when this flag is set, and a *GetScheduledPrices* command MAY also be sent.
 7902

7903 **NotificationFlags3 Attribute:** The **NotificationFlags3 attribute** shall be configured to support
 7904 the **Calendar cluster** and is implemented as a set of bit flags which are defined as shown below:

7905 **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

7906

7907 **NotificationFlags4 Attribute:** The **NotificationFlags4 attribute** shall be configured to support
 7908 the **Prepayment cluster** and is implemented as a set of bit flags which are defined as shown
 7909 below:

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

7913 **NotificationFlags5 Attribute:** The **NotificationFlags5** attribute shall be configured to support the **Device Management cluster** and is implemented as a set of bit flags which are defined as shown below:

7914

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

7915

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.

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

7950 • INTERRUPT

7951 • ARM

7952

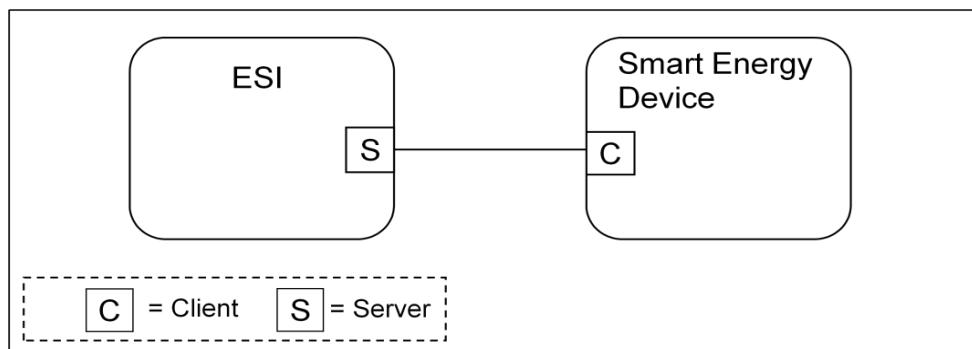
7953

7954

7955 **D.4 Price Cluster**7956 **D.4.1 Overview**

7957 The Price Cluster provides the mechanism for communicating Gas, Energy, or Water pricing
 7958 information within the premises. This pricing information is distributed to the ESI from either the
 7959 utilities or from regional energy providers. The ESI conveys the information (via the Price Cluster
 7960 mechanisms) to other Smart Energy devices.

7961



7962

7963 **Figure D-55– Price Cluster Client Server Example**

7964 Please note the ESI is defined as the Server due to its role in acting as the proxy for upstream price
 7965 management systems and subsequent data stores.

7966 **D.4.2 Server**7967 **D.4.2.1 Dependencies**

7968 Events carried using this cluster include a timestamp with the assumption that target devices
 7969 maintain a real time clock. Devices can acquire and synchronize their internal clocks via the ZCL
 7970 Time server.

7971 If a device does not support a real time clock it is assumed that the device will interpret and
 7972 utilize the “Start Now” value within the Time field.

7973 **D.4.2.2 Attributes**

7974 For convenience, the attributes defined in this cluster are arranged into sets of related
 7975 attributes; each set can contain up to 256 attributes. Attribute identifiers are encoded such that
 7976 the most significant Octet specifies the attribute set and the least significant Octet specifies the

7977 attribute within the set. The currently defined attribute sets are listed in the following Table
7978 D-73. The Price Cluster is broken down in to Delivered attribute sets 0x00 to 0x7F and Received
7979 attribute sets 0x80 to 0xFF.

7980 **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

7981

7982 **D.4.2.2.1 Tier Label (Delivered) Set**

7983

7984 **Table D-74– Tier Label Attribute Set**

Identifier	Name	Type	Length	Access	Default	Mandatory / Optional
0x0000	<i>Tier1PriceLabel</i>	Octet String	1 to 13 Octets	Read/ Write	“Tier 1”	O
0x0001	<i>Tier2PriceLabel</i>	Octet String	1 to 13 Octets	Read/ Write	“Tier 2”	O

0x0002	<i>Tier3PriceLabel</i>	Octet String	1 to 13 Octets	Read/ Write	“Tier 3”	O
0x0003	<i>Tier4PriceLabel</i>	Octet String	1 to 13 Octets	Read/ Write	“Tier 4”	O
0x0004	<i>Tier5PriceLabel</i>	Octet String	1 to 13 Octets	Read/ Write	“Tier 5”	O
0x0005	<i>Tier6PriceLabel</i>	Octet String	1 to 13 Octets	Read/ Write	“Tier 6”	O
0x0006	<i>Tier7PriceLabel</i>	Octet String	1 to 13 Octets	Read/ Write ^a	“Tier 7”	O
0x0007	<i>Tier8PriceLabel</i>	Octet String	1 to 13 Octets	Read/ Write ^a	“Tier 8”	O
0x0008	<i>Tier9PriceLabel</i>	Octet String	1 to 13 Octets	Read/ Write ^a	“Tier 9”	O
0x0009	<i>Tier10PriceLabel</i>	Octet String	1 to 13 Octets	Read/ Write ^a	“Tier 10”	O
0x000A	<i>Tier11PriceLabel</i>	Octet String	1 to 13 Octets	Read/ Write ^a	“Tier 11”	O
0x000B	<i>Tier12PriceLabel</i>	Octet String	1 to 13 Octets	Read/ Write ^a	“Tier 12”	O
0x000C	<i>Tier13PriceLabel</i>	Octet String	1 to 13 Octets	Read/ Write ^a	“Tier 13”	O
0x000D	<i>Tier14PriceLabel</i>	Octet String	1 to 13 Octets	Read/ Write ^a	“Tier 14”	O
0x000E	<i>Tier15PriceLabel</i>	Octet String	1 to 13 Octets	Read/ Write ^a	“Tier 15”	O
0x000F	<i>Tier16PriceLabel</i>	Octet String	1 to 13 Octets	Read/ Write	“Tier 16”	O
0x0010	<i>Tier17PriceLabel</i>	Octet String	1 to 13 Octets	Read/ Write	“Tier 17”	O
0x001n	<i>TierwxPriceLabel</i>	Octet String	1 to 13 Octets	Read/ Write	“Tier wx”	O
0x002n	<i>TieryzPriceLabel</i>	Octet String	1 to 13 Octets	Read/ Write	“Tier yz”	O
0x2F	<i>Tier48PriceLabel</i>	Octet String	1 to 13 Octets	Read/ Write	“Tier 48”	O
0x0030 to 0x00FF	Reserved					

7985

a. CCB 1511

7986

7987 D.4.2.2.1 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 Price* 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	<i>Block1Threshold</i>	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0101	<i>Block2Threshold</i>	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0102	<i>Block3Threshold</i>	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0103	<i>Block4Threshold</i>	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0104	<i>Block5Threshold</i>	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O

¹⁶ CCB 1511

0x0105	<i>Block6Threshold</i>	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0106	<i>Block7Threshold</i>	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0107	<i>Block8Threshold</i>	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0108	<i>Block9Threshold</i>	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0109	<i>Block10Threshold</i>	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x010A	<i>Block11Threshold</i>	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x010B	<i>Block12Threshold</i>	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x010C	<i>Block13Threshold</i>	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x010D	<i>Block14Threshold</i>	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x010E	<i>Block15Threshold</i>	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x010F	<i>BlockThresholdCount</i>	Unsigned 8-bit Integer	0x00 to 0xFF	Read Only	-	O
0x0110	<i>Tier1Block1Threshold</i>	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O

0x0111	<i>Tier1Block2Threshold</i>	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
--	--	--	--	--	--	--
0x011E	<i>Tier1Block15Threshold</i>	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x011F	<i>Tier1BlockThreshold Count</i>	Unsigned 8-bit Integer	0x00 to 0xFF	Read Only	-	O
0x0120	<i>Tier2Block1Threshold</i>	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0121	<i>Tier2Block2Threshold</i>	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
--	--	--	--	--	--	--
0x012E	<i>Tier2Block15Threshold</i>	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x012F	<i>Tier2BlockThreshold Count</i>	Unsigned 8-bit Integer	0x00 to 0xFF	Read Only	-	O
--	--	--	--	--	--	--
0x01FE	<i>Tier15Block15Threshold</i>	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x01FF	<i>Tier15BlockThreshold Count</i>	Unsigned 8-bit Integer	0x00 to 0xFF	Read Only	-	O

8005

8006 D.4.2.2.1 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

8013 fields defined in the *Publish Price* command, the formatting being defined by attributes within
 8014 the *Block Period* attribute set.

8015 **D.4.2.2.2 BlockThresholdCount Attribute**

8016 Where a single set of thresholds is used, the *BlockThresholdCount* attribute indicates the number
 8017 of applicable *BlockNThresholds*. Where more than one set of thresholds is used, each set will be
 8018 accompanied by an appropriate *TierNBlockThresholdCount* attribute (see D.4.2.2.4).

8019 **D.4.2.2.3 TierNBlockMThreshold Attributes**

8020 Attributes *Tier1Block1Threshold* through *Tier15Block15Threshold* represent the block threshold
 8021 values applicable to a specific TOU tier for a given period (typically the billing cycle). These
 8022 values may be updated by the utility on a seasonal or annual basis. The thresholds are established
 8023 such that crossing the threshold of energy consumption for the present block activates the next
 8024 higher block, which can affect the energy rate in a positive or negative manner. The values are
 8025 absolute and always increasing. The values represent threshold at the end of a block. The Unit of
 8026 Measure will be based on the fields defined in the *Publish Price* command, the formatting being
 8027 defined by attributes within the *Block Period* attribute set.

8028 **D.4.2.2.4 TierNBlockThresholdCount Attributes**

8029 The *TierNBlockThresholdCount* attributes hold the number of block thresholds applicable to a
 8030 given tier. These attributes are used in the case when a combination (TOU/Hybrid) tariff has a
 8031 separate set of thresholds for each TOU tier. Unused *TierNBlockThresholdCount* attributes shall
 8032 be set to zero.

8033 **D.4.2.2.3 Block Period (Delivered) Set**

8034 The following set of attributes provides remote access to the Price server Block Threshold
 8035 period (typically the billing cycle) information.

8036 **Table D-76– Block Period Attribute Set**

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory /Optional
0x0200	<i>StartofBlock Period</i>	UTCTime	-	Read Only	-	O
0x0201	<i>BlockPeriod Duration</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0202	<i>Threshold Multiplier</i>	Unsigned 24 bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0203	<i>Threshold Divisor</i>	Unsigned 24 bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O

0x0204	<i>BlockPeriodDurationType</i>	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**

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0300	<i>CommodityType</i>	8-bit Enumeration	0x00 to 0xFF	Read Only	-	O
0x0301	<i>Standing Charge</i>	Unsigned 32 bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x0302	<i>Conversion Factor</i>	Unsigned 32 bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	0x10000000	O
0x0303	<i>Conversion Factor TrailingDigit</i>	8-bit BitMap		Read Only	0x70	O
0x0304	<i>Calorific Value</i>	Unsigned 32 bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	0x2625A00	O
0x0305	<i>CalorificValue Unit</i>	8-bit Enumeration		Read Only	0x1	O
0x0306	<i>Calorific Value TrailingDigit</i>	8-bit BitMap		Read Only	0x60	O
0x0307 - 0x03FF	Reserved					

8067

8068

8069 D.4.2.2.4.1 CommodityType Attribute

8070 *CommodityType* provides a label for identifying the type of pricing server present. The attribute is
 8071 an enumerated value representing the commodity. The defined values are represented by the
 8072 non-mirrored values (0-127) in the *MeteringDeviceType* attribute enumerations (refer to Table
 8073 D-27).

8074 D.4.2.2.4.2 Standing Charge Attribute

8075 The value of the *Standing Charge* is a daily fixed charge associated with supplying the
 8076 commodity, measured in base unit of Currency with the decimal point located as indicated by
 8077 the Trailing Digits field of a *Publish Price* command (see sub-clause D.4.2.4.1) or
 8078 *PriceTrailingDigit* attribute (see D.4.2.2.7.7). A value of 0xFFFFFFFF indicates attribute not
 8079 used.

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/m ³
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.

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	<i>NoTierBlock1Price</i>	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x0401	<i>NoTierBlock2Price</i>	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x0402	<i>NoTierBlock3Price</i>	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x040N	<i>NoTierBlockN+1Price</i> ...	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x040F	<i>NoTierBlock16Price</i>	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x0410	<i>Tier1Block1Price</i>	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x0411	<i>Tier1Block2Price</i>	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x0412	<i>Tier1Block3Price</i>	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x041N	<i>Tier1BlockN+1Price</i> ...	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x041F	<i>Tier1Block16Price</i>	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x0420	<i>Tier2Block1Price</i>	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x042N	<i>Tier2BlockN+1Price</i> ...	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O

0x042F	<i>Tier2Block16Price</i>	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x0430	<i>Tier3Block1Price</i>	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x043N	<i>Tier3BlockN+1Price ...</i>	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x043F	<i>Tier3Block16Price</i>	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x0440	<i>Tier4Block1Price</i>	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x044N	<i>Tier4BlockN+1Price ...</i>	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x044F	<i>Tier4Block16Price</i>	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x0450	<i>Tier5Block1Price</i>	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x045N	<i>Tier5BlockN+1Price ...</i>	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x045F	<i>Tier5Block16Price</i>	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x0460	<i>Tier6Block1Price</i>	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x046N	<i>Tier6BlockN+1Price ...</i>	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x046F	<i>Tier6Block16Price</i>	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x0470	<i>Tier7Block1Price</i>	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O

0x047N	<i>Tier7BlockN+1Price ...</i>	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x047F	<i>Tier7Block16Price</i>	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x0480	<i>Tier8Block1Price</i>	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x048N	<i>Tier8BlockN+1Price ...</i>	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x048F	<i>Tier8Block16Price</i>	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x0490	<i>Tier9Block1Price</i>	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x049N	<i>Tier9BlockN+1Price ...</i>	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x049F	<i>Tier9Block16Price</i>	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x04A0	<i>Tier10Block1Price</i>	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x04AN	<i>Tier10BlockN+1Price ...</i>	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x04AF	<i>Tier10Block16Price</i>	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x04B0	<i>Tier11Block1Price</i>	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x04BN	<i>Tier11BlockN+1Price ...</i>	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x04BF	<i>Tier11Block16Price</i>	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O

0x04C0	<i>Tier12Block1Price</i>	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x04CN	<i>Tier12BlockN+1Price</i> ...	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x04CF	<i>Tier12Block16Price</i>	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x04D0	<i>Tier13Block1Price</i>	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x04DN	<i>Tier13BlockN+1Price</i> ...	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x04DF	<i>Tier13Block16Price</i>	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x04E0	<i>Tier14Block1Price</i>	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x04EN	<i>Tier14BlockN+1Price</i> ...	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x04EF	<i>Tier14Block16Price</i>	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x04F0	<i>Tier15Block1Price</i>	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x04FN	<i>Tier15BlockN+1Price</i> ...	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x04FF	<i>Tier15Block16Price</i>	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

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.

8122 D.4.2.2.6 Extended Price Information (Delivered) Set

8123 In case of TOU charging only, the price server allows support for up to 48 TOU rates. To reduce
 8124 the number of attributes, *Tier1Block1Price* through *Tier15Block1Price* attributes are reused to
 8125 represent rates for tiers 1 to 15. Rates for tiers 16 to 48 are provided in the extended price
 8126 information set.

8127 **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

8128
 8129 D.4.2.2.6.1 PriceTierN Attributes
 8130 Attributes *PriceTier16* through *PriceTier48* represent the price of Energy, Gas, or Water
 8131 delivered to the premises (i.e. delivered to the customer from the utility) at a specific price tier.

8132 D.4.2.2.6.2 CPP1 Price Attribute
 8133 Attribute *CPP1 Price* represents the price of Energy, Gas, or Water delivered to the premises (i.e.
 8134 delivered to the customer from the utility) while Critical Peak Pricing ‘CPP1’ is being applied.

8135 D.4.2.2.6.3 CPP2 Price Attribute
 8136 Attribute *CPP2 Price* represents the price of Energy, Gas, or Water delivered to the premises (i.e.
 8137 delivered to the customer from the utility) while Critical Peak Pricing ‘CPP2’ is being applied.

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	NumberofPriceTiersInUse	Unsigned 8-bit Integer	0 to 15	Read Only	0	O
0x0612	NumberofBlockThresholdsInUse	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 ₂ Unit	8-bit Enumeration		Read Only	1	O
0x0622	CO ₂ TrailingDigit	8-bit BitMap		Read Only	0	O
0x0628-0x06FF	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.

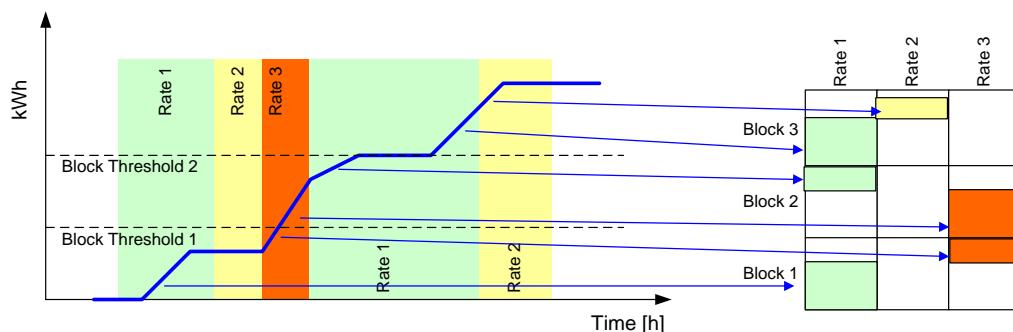
8168

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.

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 0xFF	Reserved for future use
0xFF	Not Used

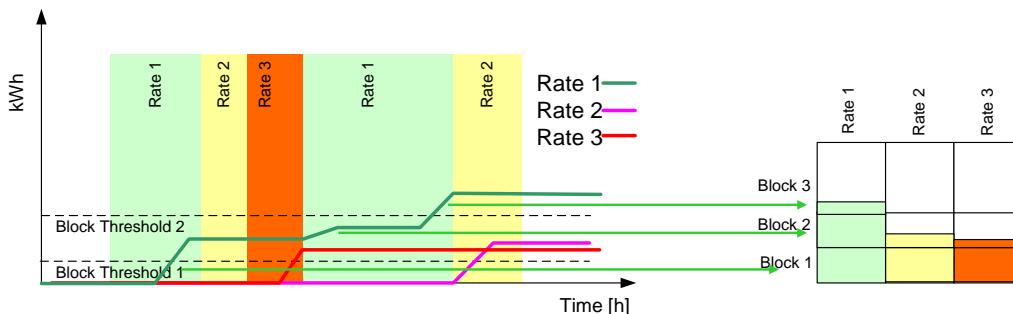
8169
8170



8171
8172

Figure D-56– Single Threshold Set applied to All Consumption

8173
8174



8175
8176

Figure D-57– Single Threshold Set applied to Each Tier Consumption

8177
8178 **Note:** Tiers 1-15 ONLY are available for hybrid Tier/Block tariffing ... Tiers 16-48 are for TOU
8179 tariffing only.
8180

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 CO₂ Attribute

8200 Used to calculate the amount of carbon dioxide (CO₂) 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 CO₂ (0.185 x 1000 kWh). The *CO₂* attribute represents the current active value.

8203 D.4.2.2.7.10 CO₂Unit Attribute

8204 This attribute is an 8-bit enumeration which defines the unit for the CO₂ attribute. The values and
 8205 descriptions for this attribute are listed in Table D-83 below. The *CO₂Unit* attribute represents
 8206 the current active value.

8207

Table D-83– CO₂Unit 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

8208

8209 D.4.2.2.7.11 *CO₂TrailingDigit* Attribute

8210 An 8-bit Bit-Map used to determine where the decimal point is located in the CO₂ 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 *CO₂TrailingDigit* 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	<i>CurrentBillingPeriodStart</i>	UTCTime	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x0701	<i>CurrentBillingPeriodDuration</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read only	-	O
0x0702	<i>LastBillingPeriodStart</i>	UTCTime	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x0703	<i>LastBillingPeriodDuration</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	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.

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 - + 0x7FFFFFFF	Read Only	0	O
0x0803 – 0x0809	Reserved					
0x080A	PaymentDiscount	Signed 32-bit Integer	- 0x7FFFFFFF - + 0x7FFFFFFF	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					

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

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 Attribute8252 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 Attribute8255 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 Attribute8260 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 Attribute8264 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 Attribute8267 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

Table D-88– Received Tier Label Attribute Set

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.

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 – 0xFFFFFFF	Read Only	-	O
0x8202	ReceivedThresholdMultiplier	Unsigned 24 bit Integer	0x000000 – 0xFFFFFFF	Read Only	-	O
0x8203	ReceivedThresholdDivisor	Unsigned 24 bit Integer	0x000000 – 0xFFFFFFF	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.4.2.2.12.4**ReceivedThresholdDivisor 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****Table D-91—Received Block Price Attribute Set**

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

0x852F	ReceivedPriceTier48	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x8530-0x85FF	Reserved					

8316 D.4.2.2.14.1 ReceivedPriceTierN Attributes8317 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 ₂ Unit	8-bit Enumeration		Read Only	1	O
0x8627	ReceivedCO ₂ TrailingDigit	8-bit BitMap		Read Only	0	O
0x8628-0x86FF	Reserved					

8323

- 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 ReceivedCO₂ Attribute
 8340 The format and use of this attribute is the same as for the ‘Delivered’ *CO₂* attribute defined in
 8341 D.4.2.2.7.9.
- 8342 D.4.2.2.15.7 ReceivedCO₂Unit Attribute
 8343 The format and use of this attribute is the same as for the ‘Delivered’ *CO₂Unit* attribute defined
 8344 in D.4.2.2.7.10.
- 8345 D.4.2.2.15.8 ReceivedCO₂TrailingDigit Attribute
 8346 The format and use of this attribute is the same as for the ‘Delivered’ *CO₂TrailingDigit* 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.

Table D-94– Received Billing Information Attribute Set

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x8700	ReceivedCurrentBillingPeriodStart	UTC Time	0x00000000 to 0xFFFFFFFF	Read Only	-	O

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x8701	ReceivedCurrentBillingPeriod Duration	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x8702	ReceivedLastBillingPeriodStart	UTC Time	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x8703	ReceivedLastBillingPeriodDuration	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x8704	ReceivedLastBillingPeriod ConsolidatedBill	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read Only	-	O
0x8705-0x87FF	Reserved					

8352

8353 D.4.2.2.16.1 ReceivedCurrentBillingPeriodStart Attribute8354 The format and use of this attribute is the same as for the ‘Delivered’ *CurrentBillingPeriodStart* attribute defined in D.4.2.2.8.1.8355 D.4.2.2.16.2 ReceivedCurrentBillingPeriodDuration Attribute8356 The format and use of this attribute is the same as for the ‘Delivered’ *CurrentBillingPeriodDuration* attribute defined in D.4.2.2.8.2.8357 D.4.2.2.16.3 ReceivedLastBillingPeriodStart Attribute8358 The format and use of this attribute is the same as for the ‘Delivered’ *LastBillingPeriodStart* attribute defined in D.4.2.2.8.3.8359 D.4.2.2.16.4 ReceivedLastBillingPeriodDuration Attribute8360 The format and use of this attribute is the same as for the ‘Delivered’ *LastBillingPeriodDuration* attribute defined in D.4.2.2.8.4.8361 D.4.2.2.16.5 ReceivedLastBillingPeriodConsolidatedBill Attribute8362 The format and use of this attribute is the same as for the ‘Delivered’ *LastBillingPeriodConsolidatedBill* attribute defined in D.4.2.2.8.5.

8363

8364 D.4.2.3 Commands Received

8365 The server side of the Price cluster is capable of receiving the commands listed in Table D-95.

8366 Table D-95– Received Command IDs for the Price Cluster

Command Identifier Field Value	Description	Mandatory / Optional
0x00	<i>GetCurrentPrice</i>	M
0x01	<i>GetScheduledPrices</i>	O
0x02	<i>PriceAcknowledgement</i>	M - Mandatory for 1.1 and later devices
0x03	<i>GetBlockPeriod(s)</i>	O
0x04	<i>GetConversionFactor</i>	O
0x05	<i>GetCalorificValue</i>	O
0x06	<i>GetTariffInformation</i>	O
0x07	<i>GetPriceMatrix</i>	O
0x08	<i>GetBlockThresholds</i>	O
0x09	<i>GetCO₂Value</i>	O
0x0A	<i>GetTierLabels</i>	O
0x0B	<i>GetBillingPeriod</i>	O
0x0C	<i>GetConsolidatedBill</i>	O
0x0D	<i>CPPEventResponse</i>	O
0x0E	<i>GetCreditPayment</i>	O
0x0F	<i>GetCurrencyConversion</i>	O
0x10	<i>GetTariffCancellation</i>	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

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
--------	---	---

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

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

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

8485 or after the stated *Earliest Start Time*. If more than one instance is requested, the active and
 8486 scheduled instances shall be sent with ascending ordered StartTime.

8487 **Min. Issuer Event ID (mandatory):** A 32-bit integer representing the minimum *Issuer Event ID*
 8488 of values to be returned by the corresponding *PublishCalorificValue* command. A value of
 8489 0xFFFFFFFF means not specified; the server shall return values irrespective of the value of the
 8490 *Issuer Event ID*.

8491 **Number of Commands (mandatory):** An 8-bit integer which represents the maximum number
 8492 of *PublishConversionFactor* commands that the client is willing to receive in response to this
 8493 command. A value of 0 would indicate all available *PublishConversionFactor* commands shall
 8494 be returned.

8495 D.4.2.3.6.3 Effect on Receipt

8496 A ZCL Default response with status NOT_FOUND shall be returned if there are no conversion
 8497 factor updates available.

8498

8499 D.4.2.3.7 GetCalorificValue Command

8500 This command initiates a *PublishCalorificValue* command(s) for scheduled calorific value
 8501 updates. A server device shall be capable of storing at least two instances, the current and (if
 8502 available) next instance to be activated in the future.

8503 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)

8504 **Figure D-64– GetCalorificValue Command Payload**

8505 D.4.2.3.7.2 Payload Details

8506 **Earliest Start Time (mandatory):** UTC Timestamp indicating the earliest start time of values to
 8507 be returned by the corresponding *PublishCalorificValue* command. The first returned
 8508 *PublishCalorificValue* command shall be the instance which is active or becomes active at or
 8509 after the stated *Earliest Start Time*. If more than one instance is requested, the active and
 8510 scheduled instances shall be sent with ascending ordered Start Time.

8511 **Min. Issuer Event ID (mandatory):** A 32-bit integer representing the minimum *Issuer Event ID*
 8512 of values to be returned by the corresponding *PublishCalorificValue* command. A value of

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 *GetBlockThresholds* 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*.

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*.

8544 **Number of Commands (mandatory):** An 8-bit integer which represents the maximum number
8545 of *PublishTariffInformation* commands that the client is willing to receive in response to this
8546 command. A value of 0 would indicate all available *PublishTariffInformation* commands shall be
8547 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.

8552 D.4.2.3.8.3 Effect on Receipt

8553 A ZCL Default response with status NOT_FOUND shall be returned if there are no tariff
8554 updates available.

8555

8556 D.4.2.3.9 GetPriceMatrix Command

8557 This command initiates a *PublishPriceMatrix* command for the scheduled Price Matrix updates.
8558 A server device shall be capable of storing at least **two** instances, current and next instance to be
8559 activated in the future.

8560 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

8562 D.4.2.3.9.2 Payload Details

8563 **Issuer Tariff ID (mandatory):** *IssuerTariffID* indicates the tariff to which the requested Price
8564 Matrix belongs.

Note: A Price Matrix instance may require multiple *PublishPriceMatrix* commands to be transmitted to the client device.

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 **GetCO₂Value Command**

8587 This command initiates *PublishCO₂Value* command(s) for scheduled CO₂ 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– GetCO₂Value Command Payload**

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 *PublishCO₂Value* command. The first returned
 8595 *PublishCO₂Value* 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 *PublishCO₂Value* 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 *PublishCO₂Value* commands that the client is willing to receive in response to this command.
 8604 A value of 0 would indicate all available *PublishCO₂Value* 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 CO₂ 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 CO₂
 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**

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.

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)

Figure D-71– GetConsolidatedBill Command Payload

8657 D.4.2.3.14.2 Payload Details

8658 **Earliest Start Time (mandatory):** UTC Timestamp indicating the earliest start time of billing information to be returned by the corresponding *PublishConsolidatedBill* command. The first returned *PublishConsolidatedBill* command shall be the instance which is active or becomes active at or after the stated *EarliestStartTime*. If more than one instance is requested, the active and scheduled instances shall be sent with ascending ordered *StartTime*.

8659 **Min. Issuer Event ID (mandatory):** A 32-bit integer representing the minimum *Issuer Event ID* of billing information to be returned by the corresponding *PublishConsolidatedBill* command. A value of 0xFFFFFFFF means not specified; the server shall return information irrespective of the value of the *Issuer Event ID*.

8660 **Number of Commands (mandatory):** An 8 bit Integer which indicates the maximum number of *PublishConsolidatedBill* commands that can be sent. A value of 0 would indicate all available *PublishConsolidatedBill* commands shall be returned.

8661 **Tariff Type (Optional):** 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). A value of 0xFF means not specified. If the *TariffType* is not specified, the server shall return all billing information regardless of tariff type. The most significant nibble is 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.

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.

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	<i>Publish Price</i>	M
0x01	<i>Publish Block Period</i>	O
0x02	<i>Publish Conversion Factor</i>	O
0x03	<i>Publish Calorific Value</i>	O
0x04	<i>PublishTariffInformation</i>	O
0x05	<i>PublishPriceMatrix</i>	O

0x06	<i>PublishBlockThresholds</i>	O
0x07	<i>PublishCO₂Value</i>	O
0x08	<i>PublishTierLabels</i>	O
0x09	<i>PublishBillingPeriod</i>	O
0x0A	<i>PublishConsolidatedBill</i>	O
0x0B	<i>PublishCPPEvent</i>	O
0x0C	<i>PublishCreditPayment</i>	O
0x0D	<i>PublishCurrencyConversion</i>	O
0x0E	<i>CancelTariff</i>	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* command (see sub-clause D.4.2.3.2), in response to a *Get Scheduled Prices* command (see sub-clause D.4.2.3.3), and when an update to the pricing information is available from the commodity provider, either before or when a TOU price becomes active. Additionally the Publish 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.

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.

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.

8796 **Current Time (mandatory):** A UTCTime field containing the current time as determined by
8797 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.: \$).

8805 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

Enumerated Value	Price Tier
0x0	No Tier Related
0x1	Reference <i>Tier1PriceLabel</i>
0x2	Reference <i>Tier2PriceLabel</i>
0x3	Reference <i>Tier3PriceLabel</i>
0x4	Reference <i>Tier4PriceLabel</i>
0x5	Reference <i>Tier5PriceLabel</i>
0x6	Reference <i>Tier6PriceLabel</i>
0x7	Reference <i>Tier7PriceLabel</i>
0x8	Reference <i>Tier8PriceLabel</i>
0x9	Reference <i>Tier9PriceLabel</i>
0xA	Reference <i>Tier10PriceLabel</i>
0xB	Reference <i>Tier11PriceLabel</i>
0xC	Reference <i>Tier12PriceLabel</i>
0xD	Reference <i>Tier13PriceLabel</i>
0xE	Reference <i>Tier14PriceLabel</i>
0xF	Dependant on the value of the <i>Extended Price Tier</i> 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

8838 *CurrentTierNSummationDelivered* attribute currently accumulating usage information by the
 8839 sum of the values of the *Register Tier* sub-field and the *Extended Register Tier* field.

8840 Both attributes can be used to calculate and display usage and subsequent costs.

8841

8842 **Table D-98– Register Tier Sub-field Enumerations**

Enumerated Value	Register Tier
0x0	No Tier Related
0x1	Usage accumulating in <i>CurrentTier1SummationDelivered</i> attribute
0x2	Usage accumulating in <i>CurrentTier2SummationDelivered</i> attribute
0x3	Usage accumulating in <i>CurrentTier3SummationDelivered</i> attribute
0x4	Usage accumulating in <i>CurrentTier4SummationDelivered</i> attribute
0x5	Usage accumulating in <i>CurrentTier5SummationDelivered</i> attribute
0x6	Usage accumulating in <i>CurrentTier6SummationDelivered</i> attribute
0x7	Usage accumulating in <i>CurrentTier7SummationDelivered</i> attribute
0x8	Usage accumulating in <i>CurrentTier8SummationDelivered</i> attribute
0x9	Usage accumulating in <i>CurrentTier9SummationDelivered</i> attribute
0xA	Usage accumulating in <i>CurrentTier10SummationDelivered</i> attribute
0xB	Usage accumulating in <i>CurrentTier11SummationDelivered</i> attribute
0xC	Usage accumulating in <i>CurrentTier12SummationDelivered</i> attribute
0xD	Usage accumulating in <i>CurrentTier13SummationDelivered</i> attribute
0xE	Usage accumulating in <i>CurrentTier14SummationDelivered</i> attribute
0xF	Dependant on the value of the <i>Extended Register Tier</i> field

8843

8844 **Start Time (mandatory):** A UTCTime field to denote the time at which the price signal becomes
 8845 valid. A Start Time of 0x00000000 is a special time denoting “now.”

8846 If the device would send a price with a Start Time of now, adjust the Duration In Minutes field
 8847 to correspond to the remainder of the price.

8848 **Duration In Minutes (mandatory):** An unsigned 16-bit field used to denote the amount of time
 8849 in minutes after the Start Time during which the price signal is valid. Maximum value means
 8850 “until changed”. If Block Charging only is in use (see sub-clause D.4.4.3 for further details), the
 8851 Duration in Minutes field of the Publish Price command shall be set to 0xFFFF indicating the
 8852 price is valid “until changed”.

8853 **Price (mandatory):** An unsigned 32-bit field containing the price of the commodity
 8854 measured in base unit of Currency per Unit of Measure with the decimal point located as
 8855 indicated by the Price Trailing Digit field when the commodity is delivered to the premises.

8856 **Price Ratio (optional):** An unsigned 8-bit field that gives the ratio of the price denoted in the
 8857 Price field to the “normal” price chosen by the commodity provider. This field is thought
 8858 to be useful in situations where client devices may simply be interested in pricing levels or
 8859 ratios. The value in this field should be scaled by a factor of 0.1, giving a range of ratios from 0.1
 8860 to 25.4. A value of 0xFF indicates the field is not used and 0x00 is an invalid value.

8861 **Generation Price (optional):** An unsigned 32-bit field containing the price of the commodity
 8862 measured in base unit of Currency per Unit of Measure with the decimal point located as indicated
 8863 by the Price Trailing Digit field when the commodity is received from the premises. An example
 8864 use of this field is in energy markets where the price of electricity from the grid is different than
 8865 the price of electricity placed on the grid. A value of 0xFFFFFFFF indicates the field is not used.

8866 **Generation Price Ratio (optional):** An unsigned 8-bit field that gives the ratio of the price
 8867 denoted in the Generation Price field to the “normal” price chosen by the commodity provider.
 8868 This field is thought to be useful in situations where client devices may simply be interested in
 8869 pricing levels or ratios. The value in this field should be scaled by a factor of 0.1, giving a
 8870 range of ratios from 0.1 to 25.4 A value of 0xFF indicates the field is not used and 0x00 is an
 8871 invalid value.

8872 **Alternate Cost Delivered (optional):** An unsigned 32-bit Integer field that provides a
 8873 mechanism to describe an alternative measure of the cost of the energy consumed. An
 8874 example of an Alternate Cost might be the emissions of CO₂ for each kWh of electricity
 8875 consumed providing a measure of the environmental cost. Another example is the emissions of
 8876 CO₂ for each cubic meter of gas consumed (for gas metering). A different value for each price
 8877 tier may be provided which can be used to reflect the different mix of generation that is
 8878 associated with different TOU rates. A value of 0xFFFFFFFF indicates the field is not used.

8879 **Alternate Cost Unit (optional):** An 8-bit enumeration identifying the unit (as specified in
 8880 Table D-99) for the Alternate Cost Delivered field. A value of 0xFF indicates the field is not
 8881 used.

8882 **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

8883

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.

8899 If Bit 1 is set, this indicates that the total number of tiers exceeds the 15 specified in the
8900 command; this shall indicate to a client complying with this specification that it should read the
8901 total number of tiers using the *GetTariffInformation* command.

8902 The BitMap for this field is described in Table D-100.

Table D-100— Price Control Field BitMap

Table D-100 Price Control Field Breakup	
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

8904
8905 **Number of Generation Tiers (optional):** Specifies the total number of generation tiers
8906 applicable in the current tariff, valid values are 0-48.

8907 **Generation Tier (optional):** An 8-bit enumerated value specifying the current generation tier.
8908 See Table D-101.

Table D-101 Generation Tier Enumerations

Enumerated Value	Description
0x00	Reserved
0x01	Usage accumulating in <i>CurrentTier1SummationReceived</i> attribute
0x02	Usage accumulating in <i>CurrentTier2SummationReceived</i> attribute
0x03	Usage accumulating in <i>CurrentTier3SummationReceived</i> attribute
0x04	Usage accumulating in <i>CurrentTier4SummationReceived</i> attribute
0x05	Usage accumulating in <i>CurrentTier5SummationReceived</i> attribute
0x06	Usage accumulating in <i>CurrentTier6SummationReceived</i> attribute
0x07	Usage accumulating in <i>CurrentTier7SummationReceived</i> attribute
0x08	Usage accumulating in <i>CurrentTier8SummationReceived</i> attribute
0x09	Usage accumulating in <i>CurrentTier9SummationReceived</i> attribute
0x0A	Usage accumulating in <i>CurrentTier10SummationReceived</i> attribute
0x0B	Usage accumulating in <i>CurrentTier11SummationReceived</i> attribute
0x0C	Usage accumulating in <i>CurrentTier12SummationReceived</i> attribute
0x0D	Usage accumulating in <i>CurrentTier13SummationReceived</i> attribute
0x0E	Usage accumulating in <i>CurrentTier14SummationReceived</i> attribute
0x0F	Usage accumulating in <i>CurrentTier15SummationReceived</i> attribute
0x10	Usage accumulating in <i>CurrentTier16SummationReceived</i> attribute
0x11	Usage accumulating in <i>CurrentTier17SummationReceived</i> attribute
0x12	Usage accumulating in <i>CurrentTier18SummationReceived</i> attribute
0x13	Usage accumulating in <i>CurrentTier19SummationReceived</i> attribute
0x14	Usage accumulating in <i>CurrentTier20SummationReceived</i> attribute
0x15	Usage accumulating in <i>CurrentTier21SummationReceived</i> attribute
0x16	Usage accumulating in <i>CurrentTier22SummationReceived</i> attribute
0x17	Usage accumulating in <i>CurrentTier23SummationReceived</i> attribute
0x18	Usage accumulating in <i>CurrentTier24SummationReceived</i> attribute
0x19	Usage accumulating in <i>CurrentTier25SummationReceived</i> attribute
0x1A	Usage accumulating in <i>CurrentTier26SummationReceived</i> attribute
0x1B	Usage accumulating in <i>CurrentTier27SummationReceived</i> attribute
0x1C	Usage accumulating in <i>CurrentTier28SummationReceived</i> attribute
0x1D	Usage accumulating in <i>CurrentTier29SummationReceived</i> attribute
0x1E	Usage accumulating in <i>CurrentTier30SummationReceived</i> attribute
0x1F	Usage accumulating in <i>CurrentTier31SummationReceived</i> attribute
0x20	Usage accumulating in <i>CurrentTier32SummationReceived</i> attribute
0x21	Usage accumulating in <i>CurrentTier33SummationReceived</i> attribute
0x22	Usage accumulating in <i>CurrentTier34SummationReceived</i> attribute
0x23	Usage accumulating in <i>CurrentTier35SummationReceived</i> attribute
0x24	Usage accumulating in <i>CurrentTier36SummationReceived</i> attribute
0x25	Usage accumulating in <i>CurrentTier37SummationReceived</i> attribute
0x26	Usage accumulating in <i>CurrentTier38SummationReceived</i> attribute

8910	0x27	Usage accumulating in <i>CurrentTier39SummationReceived</i> attribute
8911	0x28	Usage accumulating in <i>CurrentTier40SummationReceived</i> attribute
	0x29	Usage accumulating in <i>CurrentTier41SummationReceived</i> attribute
	0x2A	Usage accumulating in <i>CurrentTier42SummationReceived</i> attribute
	0x2B	Usage accumulating in <i>CurrentTier43SummationReceived</i> attribute
	0x2C	Usage accumulating in <i>CurrentTier44SummationReceived</i> attribute
	0x2D	Usage accumulating in <i>CurrentTier45SummationReceived</i> attribute
	0x2E	Usage accumulating in <i>CurrentTier46SummationReceived</i> attribute
	0x2F	Usage accumulating in <i>CurrentTier47SummationReceived</i> attribute
	0x30	Usage accumulating in <i>CurrentTier48SummationReceived</i> attribute
	0x31 – 0xFF	Reserved

8912 **Extended Number of Price Tiers (optional):** Where the maximum number of price tiers
8913 available exceeds the value of 15 supported by the *Number of Price Tiers* sub-field, this
8914 enumerated field is used in conjunction with the *Number of Price Tiers* sub-field to indicate the
8915 maximum number of price tiers available. Valid values for the *Extended Number of Price Tiers*
8916 field are from 1 to 33, indicating a maximum number of tiers available from 16 to 48
8917 respectively. A value of zero indicates that the maximum number of price tiers available is
8918 indicated by the *Number of Price Tiers* sub-field alone.

8919 **Extended Price Tier (optional):** Where the current Price Tier exceeds the value of 15 supported
8920 by the *Price Tier* sub-field, this enumerated field is used in conjunction with the *Price Tier* sub-
8921 field to indicate the current Price Tier. Valid values for the *Extended Price Tier* field are from 1
8922 to 33, indicating a current Price Tier of 16 to 48 respectively as shown in Table D-102. A value
8923 of zero indicates that the current status of the Price Tier is indicated by the *Price Tier* sub-field
8924 alone.

8925 **Table D-102– Extended Price Tier Field Enumerations**

Enumerated Value	Price Tier
0x00	Refer to <i>Price Tier</i> sub-field
0x01	Reference <i>Tier16PriceLabel</i>
0x02	Reference <i>Tier17PriceLabel</i>
0x03	Reference <i>Tier18PriceLabel</i>
0x04	Reference <i>Tier19PriceLabel</i>
0x05	Reference <i>Tier20PriceLabel</i>
0x06	Reference <i>Tier21PriceLabel</i>
0x07	Reference <i>Tier22PriceLabel</i>
0x08	Reference <i>Tier23PriceLabel</i>
0x09	Reference <i>Tier24PriceLabel</i>
0x0A	Reference <i>Tier25PriceLabel</i>
0x0B	Reference <i>Tier26PriceLabel</i>
0x0C	Reference <i>Tier27PriceLabel</i>
0x0D	Reference <i>Tier28PriceLabel</i>
0x0E	Reference <i>Tier29PriceLabel</i>
0x0F	Reference <i>Tier30PriceLabel</i>
0x10	Reference <i>Tier31PriceLabel</i>
0x11	Reference <i>Tier32PriceLabel</i>
0x12	Reference <i>Tier33PriceLabel</i>
0x13	Reference <i>Tier34PriceLabel</i>
0x14	Reference <i>Tier35PriceLabel</i>
0x15	Reference <i>Tier36PriceLabel</i>
0x16	Reference <i>Tier37PriceLabel</i>
0x17	Reference <i>Tier38PriceLabel</i>

0x18	Reference <i>Tier39PriceLabel</i>
0x19	Reference <i>Tier40PriceLabel</i>
0x1A	Reference <i>Tier41PriceLabel</i>
0x1B	Reference <i>Tier42PriceLabel</i>
0x1C	Reference <i>Tier43PriceLabel</i>
0x1D	Reference <i>Tier44PriceLabel</i>
0x1E	Reference <i>Tier45PriceLabel</i>
0x1F	Reference <i>Tier46PriceLabel</i>
0x20	Reference <i>Tier47PriceLabel</i>
0x21	Reference <i>Tier48PriceLabel</i>

8926

8927 **Extended Register Tier (mandatory):** Where the current Register Tier exceeds the value of 15
 8928 supported by the *Register Tier* sub-field, this enumerated field is used in conjunction with the
 8929 *Register Tier* sub-field to indicate which *CurrentTierNSummationDelivered* attribute, found in
 8930 sub-clause D.3.2.2.2, is accumulating usage information. Valid values for the *Extended Register*
 8931 *Tier* field are from 1 to 33, indicating a current Register Tier of 16 to 48 respectively as shown in
 8932 Table D-103. A value of zero indicates that the current status of the Register Tier is indicated by
 8933 the *Register Tier* sub-field alone.

8934

Table D-103– Extended Register Tier Field Enumerations

Enumerated Value	Register Tier
0x00	Refer to <i>Register Tier</i> sub-field
0x01	Usage accumulating in <i>CurrentTier16SummationDelivered</i> attribute
0x02	Usage accumulating in <i>CurrentTier17SummationDelivered</i> attribute
0x03	Usage accumulating in <i>CurrentTier18SummationDelivered</i> attribute
0x04	Usage accumulating in <i>CurrentTier19SummationDelivered</i> attribute
0x05	Usage accumulating in <i>CurrentTier20SummationDelivered</i> attribute
0x06	Usage accumulating in <i>CurrentTier21SummationDelivered</i> attribute
0x07	Usage accumulating in <i>CurrentTier22SummationDelivered</i> attribute
0x08	Usage accumulating in <i>CurrentTier23SummationDelivered</i> attribute
0x09	Usage accumulating in <i>CurrentTier24SummationDelivered</i> attribute
0x0A	Usage accumulating in <i>CurrentTier25SummationDelivered</i> attribute
0x0B	Usage accumulating in <i>CurrentTier26SummationDelivered</i> attribute
0x0C	Usage accumulating in <i>CurrentTier27SummationDelivered</i> attribute
0x0D	Usage accumulating in <i>CurrentTier28SummationDelivered</i> attribute
0x0E	Usage accumulating in <i>CurrentTier29SummationDelivered</i> attribute

0x0F	Usage accumulating in <i>CurrentTier30SummationDelivered</i> attribute
0x10	Usage accumulating in <i>CurrentTier31SummationDelivered</i> attribute
0x11	Usage accumulating in <i>CurrentTier32SummationDelivered</i> attribute
0x12	Usage accumulating in <i>CurrentTier33SummationDelivered</i> attribute
0x13	Usage accumulating in <i>CurrentTier34SummationDelivered</i> attribute
0x14	Usage accumulating in <i>CurrentTier35SummationDelivered</i> attribute
0x15	Usage accumulating in <i>CurrentTier36SummationDelivered</i> attribute
0x16	Usage accumulating in <i>CurrentTier37SummationDelivered</i> attribute
0x17	Usage accumulating in <i>CurrentTier38SummationDelivered</i> attribute
0x18	Usage accumulating in <i>CurrentTier39SummationDelivered</i> attribute
0x19	Usage accumulating in <i>CurrentTier40SummationDelivered</i> attribute
0x1A	Usage accumulating in <i>CurrentTier41SummationDelivered</i> attribute
0x1B	Usage accumulating in <i>CurrentTier42SummationDelivered</i> attribute
0x1C	Usage accumulating in <i>CurrentTier43SummationDelivered</i> attribute
0x1D	Usage accumulating in <i>CurrentTier44SummationDelivered</i> attribute
0x1E	Usage accumulating in <i>CurrentTier45SummationDelivered</i> attribute
0x1F	Usage accumulating in <i>CurrentTier46SummationDelivered</i> attribute
0x20	Usage accumulating in <i>CurrentTier47SummationDelivered</i> attribute
0x21	Usage accumulating in <i>CurrentTier48SummationDelivered</i> 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

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.

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).

8977 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.

8982 **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 (0xFFFFFFF) means 'until changed'.

8985 **Block Period Control (mandatory):** Identifies additional control options for the block period event. A value of 0x00 indicates field not used.

8987 The BitMap for this field is described in Table D-104.

8988 **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

8989 **Price Acknowledgement:** Indicates whether a *Price Acknowledgment* command shall be returned on receipt of this *Publish Block Period* command.

8992 **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.

8994 **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.

8997 Enumerated values for the Duration Timebase are shown in Table D-105:

8998 **Table D-105– Block Period DurationTimebase Enumeration**

Value	Description
0x0	Minutes (default)
0x1	Days
0x2	Weeks
0x3	Months
0x4-0xF	Reserved

8999

9000 Enumerated values for the Duration Control are shown in Table D-106:

9001

Table D-106– Block Period Duration Control Enumeration

Value	Description
0x0	Start of Timebase
0x1	End of Timebase
0x2	Not Specified
0x3-0xF	Reserved

9002 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.

9007

9008 **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.

9012 **Tariff Resolution Period (mandatory):** An 8 bit enumeration identifying the resolution period for the block tariff. See Table D-107:

9014

Table D-107– Tariff Resolution Period Enumeration

Value	Description
0x00	Not Defined
0x01	Block Period
0x02	1 Day
0x03-0xFF	Reserved

9015 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.

9017 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:

9021 *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*

9023 period. For example, if the Block Period is 90 days and the day is number 45 in the period, then
 9024 the thresholds which determine the cost to date on that day will be 50% of the thresholds defined
 9025 for the whole Block period. This creates an averaging effect on the block tariff and prevents the
 9026 customer from being exposed to one or more potentially large cost changes for many days during
 9027 the billing period which can create customer concern, particularly in prepayment applications,
 9028 and replacing these with cost changes during each day which are less apparent, but create the
 9029 same total charges.

9030

9031 **D.4.2.4.3 PublishConversionFactor Command**

9032 The *PublishConversionFactor* command is sent in response to a *GetConversionFactor* command
 9033 or if a new conversion factor is available.

9034 Clients shall be capable of storing at least two instances of the Conversion Factor, the currently
 9035 active one and the next one.

9036 **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)

9037 **Figure D-76—*PublishConversionFactor* Command Payload**

9038

9039 **D.4.2.4.3.2 Payload Details**

9040 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider.

9041 **Start Time (mandatory):** A UTCTime field to denote the time at which the value becomes valid.
 9042 The value remains valid until replaced by a newer one.

9043 **Conversion Factor (mandatory):** See Price Cluster Commodity attributes (see sub-clause
 9044 D.4.2.2.4.3).

9045 **Conversion Factor Trailing Digit (mandatory):** See Price Cluster Commodity attributes (see
 9046 sub-clause D.4.2.2.4.4).

9047

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.

9073

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)

9074

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)

9075

Figure D-78—PublishTariffInformation Command Payload

D.4.2.4.5.2 Payload Details

9077 **ProviderID (mandatory):** A unique identifier for the commodity supplier. The *ProviderID* in
 9078 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
 9079 depending on *TariffType*) except for the case where a change of supplier is pending and the new
 9080 supplier wishes to publish its tariff information in advance.

9081 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. When
 9082 new information is provided that replaces older information for the same time period, this field
 9083 allows devices to determine which information is newer. The value contained in this field is a
 9084 unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)
 9085 identifying when the Publish command was issued. Thus, newer information will have a value in
 9086 the Issuer Event ID field that is larger than older information.

9087 **Issuer Tariff ID (mandatory):** Unique identifier generated by the commodity supplier.

9088 **Start Time (mandatory):** A UTC Time field to denote the time at which the price signal
 9089 becomes valid. A start date/time of 0x00000000 shall indicate that the command should be
 9090 executed immediately.

9091 **Tariff Type/Charging Scheme (mandatory):** An 8-bit bitmap identifying the type of tariff
 9092 published in this command. The least significant nibble represents an enumeration of the tariff
 9093 type as detailed in Table D-108 (Generation Meters shall use the ‘Received’ Tariff), the most
 9094 significant nibble represents an enumeration specifying the charging scheme as detailed in Table
 9095 D-109.

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 in D.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

9119 not used. When publishing Received tariffs (according to *TariffType*) this field should be set to
9120 0xFFFFFFFF.

9121 **TierBlockMode (mandatory):** The format and use of this field is the same as for the
9122 *TierBlockMode* attribute or *ReceivedTierBlockMode* attribute (depending on *TariffType*) as
9123 defined in D.4.2.2.7.4 and D.4.2.2.15.4 respectively. In case of TOU or Block Charging only,
9124 this field is not used and shall be set to 0xFF. For combined Block/TOU charging, this field is
9125 mandatory and must be set to a valid value.

9126 **BlockThresholdMultiplier (mandatory):** *BlockThresholdMultiplier* provides a value to be
9127 multiplied against Threshold parameter(s). If present, this attribute must be applied to all Block
9128 Threshold values to derive values that can be compared against the
9129 *CurrentBlockPeriodConsumptionDelivered* attribute within the Metering cluster. This parameter
9130 must be used in conjunction with the *BlockThresholdDivisor* parameter(s). In case no multiplier
9131 is defined, this field shall be set to 1.

9132 **BlockThresholdDivisor (mandatory):** *BlockThresholdDivisor* provides a value to divide the
9133 result of applying the *ThresholdMultiplier* attribute to Block Threshold values to derive values
9134 that can be compared against the *CurrentBlockPeriodConsumptionDelivered* attribute within the
9135 Metering cluster. This attribute must be used in conjunction with the *BlockThresholdMultiplier*
9136 parameter(s). In case no divisor is defined, this field shall be set to 1.

9137 D.4.2.4.5.3 Effect on Receipt

9138 If the client is unable to store this *PublishTariffInformation* command, the device should respond
9139 using a ZCL Default Response with a status of INSUFFICIENT_SPACE.

9140

9141 D.4.2.4.6 PublishPriceMatrix Command

9142 The *PublishPriceMatrix* command is used to publish the Block Price Information Set (up to 15
9143 tiers x 15 blocks) and the Extended Price Information Set (up to 48 tiers). The
9144 *PublishPriceMatrix* command is sent in response to a *GetPriceMatrix* command.

9145 Clients should be capable of storing at least **two** instances of the Price Matrix, the currently
9146 active and the next one.

9147 There may be a separate Price Matrix for consumption delivered and received; in this case, each
9148 Price Matrix will be identified by a different *IssuerTariffId* value.

9149 The Price server shall send only the number of tiers and blocks as defined in the corresponding
9150 *PublishTariffInformation* command (NumberofPriceTiersinUse,
9151 NumberofBlockThresholdsinUse+1).

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.

9179 **Total Number of Commands (mandatory):** In the case that an entire payload does not fit into
9180 one message, the *Total Number of Commands* field indicates the total number of sub-commands
9181 in the message.

9182 **Sub-Payload Control (mandatory):** An 8-bit bitmap, the least significant bit of which specifies
9183 the information type stored in the sub payload (see Table D-110). The remaining bits are
9184 reserved.

9185 **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

9186

9187 **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)	...

9188 **Figure D-80– PriceMatrix Command Sub-Payload**

9189 **Tier/Block ID (Mandatory):** The Tier/Block ID specifies the TOU Tier or the TOU Tier and
9190 Block that the subsequent Price field in the command applies to. If Bit 0 of the *Sub-Payload*
9191 *Control* field is set to Zero, then the least significant nibble represents a value specifying the
9192 block number and the most significant nibble represents the Tier that the subsequent *Price* field
9193 applies to. Valid values for the Block Number sub-field are 0 to 15 reflecting block 1 (0) to block
9194 16(15). Valid values for the Tiers sub-field are from 0 to 15 reflecting no tiers to tier fifteen.

9195 If Bit 0 of the *Sub-Payload Control* field is set to one, then the field is an 8-bit value specifying
9196 the TOU Tier that the subsequent *Price* field applies to. Valid values are 1 to 48.

9197 **Price (Mandatory):** This field holds the price information for the Block/TOU or TOU identified
9198 by the previous *Tier/Block ID* field. The price information is provided in a base unit of *Currency*
9199 with the decimal point located as indicated by the *Trailing Digits* field of a
9200 *PublishTariffInformation* command or by the attribute defined in the *Tariff Information Attribute*
9201 *Set*.

9202 **NOTE:** The number of blocks in use is one greater than the number of block thresholds in use.
9203 For TOU charging only (number of block thresholds in use = 0, number of blocks in use = 1),
9204 the price information of block 1, tier 1 to 15 shall be used.

9205

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 active and the next one.

9210 There may be a separate set of Block Thresholds for consumption delivered and received; in this 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 (*NumberofBlockThresholdsInUse*) as defined in the *PublishTariffInformation* command.

9214 The maximum application payload may not be sufficient to transfer all thresholds in one command. In this case the Price server may send two consecutive *PublishBlockThreshold* commands (*CommandIndex* set to 0 and 1 respectively); both 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 *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 the commodity provider. This field allows differentiation in deregulated markets where multiple commodity providers may be available.

9226 **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.

9232 **Start Time (mandatory):** A UTC Time field to denote the time at which the price signal
9233 becomes valid. A start date/time of 0x00000000 shall indicate that the command should be
9234 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.

9237 Command Index (mandatory): The *Command Index* is used to count the payload fragments in
9238 the case where the entire payload does not fit into one message. The *Command Index* starts at 0
9239 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

9246 D.4.2.4.7.2.1 BlockThreshold Sub-Payload

9247 The *BlockThreshold Sub-Payload* consists of multiple sets of data which consist of a Tier ID,
9248 Block Threshold Count and the threshold values associated with the stated Tier. The number of
9249 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

9251 **Tier/NumberOfBlockThresholds:** The *Tier/NumberOfBlockThresholds* field is an 8 bitmap.
9252 The format of the bitmap is decided by bit0 of the sub-payload control field.

9253 If Bit0 of the *Sub-Payload Control* field is 0, then the least significant nibble represents a value
 9254 specifying the number of thresholds to follow in the command. The most significant nibble
 9255 represents the Tier that the subsequent block threshold values apply to.

9256 If Bit0 of the *Sub-Payload Control* field is 1, then the most significant nibble is unused and
 9257 should be set to 0.

9258 Valid values for the *NumberOfBlockThresholds* sub-field are 0 to 15 reflecting no block in use
 9259 (0) to block 15(15).Valid values for the *Tiers* sub-field are from 0 to 15 reflecting no tier to tier
 9260 fifteen.

9261 If the thresholds for a particular tier (Bit0 of the *Sub-Payload Control* field is 0) or the total
 9262 number of thresholds (Bit0 of the *Sub-Payload Control* field is 1) will not fit into a single
 9263 *PublishBlockThresholds* command, then the value of this *NumberOfBlockThresholds* sub-field
 9264 shall indicate the number of thresholds of the relevant type contained within this particular
 9265 command only.

9266 **BlockThreshold:** The Block Thresholds represent the threshold values applicable to an
 9267 individual block period and, where applicable, to a particular tier.

9268 The thresholds are established such that crossing the threshold of energy consumption for the
 9269 present block activates the next higher block, which can affect the energy rate in a positive or
 9270 negative manner. The values are absolute and always increasing. The values represent the
 9271 threshold at the end of a block. The Unit of Measure will be based on the fields defined in the
 9272 *PublishTariffInformation* command, the formatting being defined by *ThresholdDivisor* and
 9273 *ThresholdMultiplier*.

9274

9275 D.4.2.4.8 **PublishCO₂Value Command**

9276 The *PublishCO₂Value* command is sent in response to a *GetCO₂Value* command or if a new CO₂
 9277 conversion factor is available.

9278 Clients should be capable of storing at least **two** instances of the CO₂ conversion factor, the
 9279 currently active and the next one.

9280 D.4.2.4.8.1 **Payload Format**

9281 The *PublishCO₂Value* 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

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– *PublishCO₂Value* Command Payload

- 9282 **D.4.2.4.8.2 Payload Details**
- 9283 **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.
- 9284 **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.
- 9285 **Start Time (mandatory):** A UTC Time field to denote the time at which the CO₂ 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 *PublishCO₂Value* 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 *PublishCO₂Value* command is used in place of this cancellation mechanism).
- 9286 **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.
- 9287 **CO₂ Value (mandatory):** The format and use of this field is the same as for the CO₂ attribute or ReceivedCO₂ 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.
- 9288 **CO₂ Unit (mandatory):** The format and use of this field is the same as for the CO₂Unit attribute or ReceivedCO₂Unit 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.
- 9289 **CO₂ Trailing Digit (mandatory):** The format and use of this field is the same as for the CO₂TrailingDigit attribute or ReceivedCO₂TrailingDigit 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.

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.

9335 **Total Number of Commands (mandatory):** In the case where the entire payload does not fit
9336 into one message, the *Total Number of Commands* field indicates the total number of sub-
9337 commands in the message.

9338 **Number of Labels (mandatory):** The number of *Tier ID/Tier Label* sets contained within the
9339 command.

9340 **Tier ID (mandatory):** The tier number that the associated *Tier Label* applies to.

9341 **Tier Label (mandatory):** ZCL Octet String field capable of storing a 12 character string (the
9342 first character indicates the string length, represented in hexadecimal format) encoded in the
9343 UTF-8 format.

9344

9345 **D.4.2.4.10 PublishBillingPeriod Command**

9346 The *PublishBillingPeriod* command is generated in response to receiving a *GetBillingPeriod(s)*
9347 command or when an update to the Billing schedule is available from the commodity supplier.

9348 Nested and overlapping *PublishBillingPeriod* commands are not allowed. In the case of
9349 overlapping billing periods, the period with the newer *IssuerEventID* takes priority over all
9350 nested and overlapping periods. All existing periods that overlap, even partially, should be
9351 removed. Note however that there may be separate billing schedules for consumption delivered
9352 and received.

9353 **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)

9354 **Figure D-85– *PublishBillingPeriod* Command Payload**

9355 **D.4.2.4.10.2 Payload Details**

9356 **Provider ID (mandatory):** An unsigned 32-bit field containing a unique identifier for
9357 the commodity provider. This field allows differentiation in deregulated markets where multiple
9358 commodity providers may be available.

9359 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. When
9360 new information is provided that replaces older information for the same time period, this field
9361 allows devices to determine which information is newer. The value contained in this field is a
9362 unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)

9363 identifying when the Publish command was issued. Thus, newer information will have a value in
9364 the *Issuer Event ID* field that is larger than older information.

9365 **Billing Period Start Time (mandatory):** A UTC Time field to denote the time at which the
9366 billing period starts. A start time of 0x00000000 is a special time denoting “now”. A start
9367 date/time of 0xFFFFFFFF shall cause an existing *PublishBillingPeriod* command with the same
9368 *Provider ID* and *Issuer Event ID* to be cancelled (note that, in markets where permanently active
9369 price information is required for billing purposes, it is recommended that a
9370 replacement/superseding *PublishBillingPeriod* command is used in place of this cancellation
9371 mechanism).

9372 **Billing Period Duration (mandatory):** An unsigned 24-bit field to denote the billing period
9373 duration. The duration units are defined by the *Billing Period Duration Type* field.

9374 Billing periods are always repeating, i.e. after *BillingPeriodDuration* has elapsed since a
9375 *BillingPeriodStartTime*, a new billing period will start with the same duration.

9376 **Billing Period Duration Type (mandatory):** An 8-bit bitmap where the least significant nibble
9377 is an enumerated sub-field indicating the time base used for the duration, and the most significant
9378 nibble is an enumerated sub-field providing duration control. Enumerated values for the Duration
9379 Timebase are shown in Table D-105. Enumerated values for the Duration Control are shown in
9380 Table D-106.

9381 **Tariff Type (mandatory):** An 8-bit bitmap identifying the type of tariff published in this
9382 command. The least significant nibble represents an enumeration of the tariff type as detailed in
9383 Table D-108 (Generation Meters shall use the ‘Received’ Tariff). The most significant nibble is
9384 reserved.

9385

9386 **D.4.2.4.11 PublishConsolidatedBill Command**

9387 The *PublishConsolidatedBill* command is used to make consolidated billing information from
9388 previous billing periods available to other end devices. This command is issued in response to a
9389 *GetConsolidatedBill* command or if new billing information is available.

9390 Nested and overlapping *PublishConsolidatedBill* commands are not allowed. In the case of
9391 overlapping consolidated bills, the bill with the newer *IssuerEventID* takes priority over all
9392 nested and overlapping bills. All existing bills that overlap, even partially, should be removed.
9393 Note however that there may be separate consolidated bills for consumption delivered and
9394 received.

9395 A server device shall be capable of storing **five** consolidated bill command events as a minimum.

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 the commodity provider. This field allows differentiation in deregulated markets where multiple commodity providers may be available.

9400

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.

9401

Billing Period Start Time (mandatory): A UTC Time field containing the start time of the related billing period. 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 *PublishConsolidatedBill* 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 *PublishConsolidatedBill* command is used in place of this cancellation mechanism).

9402

Billing Period Duration (mandatory): An unsigned 24-bit field denoting the duration of the related billing period. The duration units are defined by the *Billing Period Duration Type* field.

9403

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.

9404

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

9424 Table D-108 (Generation Meters shall use the ‘Received’ Tariff). The most significant nibble is
 9425 reserved.

9426 **Consolidated Bill (mandatory):** An unsigned 32-bit field containing the consolidated bill value
 9427 for the stated billing period. The *Consolidated Bill* field should be provided in the same currency
 9428 as used in the Price cluster.

9429 **Currency (mandatory):** An unsigned 16-bit field containing identifying information concerning
 9430 the local unit of currency used in the *Consolidated Bill* field.

9431 The value of the currency field should match the values defined by ISO 4217.

9432 **BillTrailingDigit (mandatory):** An 8-bit field used to determine where the decimal point is
 9433 located in the *Consolidated Bill* field. The most significant nibble contains the *Trailing Digit* sub-
 9434 field which indicates the number of digits to the right of the decimal point.

9435

9436 **D.4.2.4.12 PublishCPPEvent Command**

9437 *Note: The PublishCPPEvent command in this revision of this specification is provisional and
 9438 not certifiable. This feature may change before reaching certifiable status in a future revision of
 9439 this specification.*

9440 The *PublishCPPEvent* command is sent from an ESI to its Price clients to notify them of a
 9441 Critical Peak Pricing (CPP) event.

9442 **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)

9443 **Figure D-87– PublishCPPEvent Command Payload**

9444 **D.4.2.4.12.2 Payload Details**

9445 **Provider ID (mandatory):** An unsigned 32-bit field containing a unique identifier for
 9446 the commodity provider. This field allows differentiation in deregulated markets where multiple
 9447 commodity providers may be available.

9448 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. When
 9449 new information is provided that replaces older information for the same time period, this field

9450 allows devices to determine which information is newer. The value contained in this field is a
 9451 unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)
 9452 identifying when the Publish command was issued. Thus, newer information will have a value in
 9453 the *Issuer Event ID* field that is larger than older information.

9454 **Start Time (mandatory):** A UTC Time field to denote the time at which the CPP event begins.
 9455 A start date/time of 0x00000000 shall indicate that the command should be executed
 9456 immediately. A start date/time of 0xFFFFFFFF shall cause an existing *PublishCPPEvent*
 9457 command with the same *Provider ID* and *Issuer Event ID* to be cancelled (note that, in markets
 9458 where permanently active price information is required for billing purposes, it is recommended
 9459 that a replacement/superseding *PublishCPPEvent* command is used in place of this cancellation
 9460 mechanism).

9461 **Duration in Minutes:** Defines the duration of the CPP event.

9462 **Tariff Type (mandatory):** An 8-bit bitmap identifying the type of tariff published in this
 9463 command. The least significant nibble represents an enumeration of the tariff type as detailed in
 9464 Table D-108 (Generation Meters shall use the ‘Received’ Tariff). The most significant nibble is
 9465 reserved.

9466 **CPP Price Tier (mandatory):** An 8-bit enumeration identifying the price tier associated with
 9467 this CPP event. The price(s) contained in the active price matrix for that price tier will override
 9468 the normal pricing scheme. Prices ‘CPP1’ and ‘CPP2’ are reserved for this purposes (see
 9469 D.4.2.2.6 for further details).

9470 **Table D-112– CPP Auth Enumeration**

Value	Description
0	‘CPP1’
1	‘CPP2’
2 - 255	Reserved

9471

9472 **CPP Auth (mandatory):** An 8-bit enumeration identifying the status of the CPP event:

9473 **Table D-113– CPP Auth Enumeration**

Value	Description
0	Pending
1	Accepted
2	Rejected
3	Forced
4 - 255	Reserved

9474

9475 D.4.2.4.12.3 When Generated

9476 The *PublishCPPEvent* command is generated when the energy provider has requested the consumer to accept a CPP, when the consumer has accepted the CPP, or if the ESI has received a
 9477 *CPPEventResponse* command. See Figure D-88.

9479

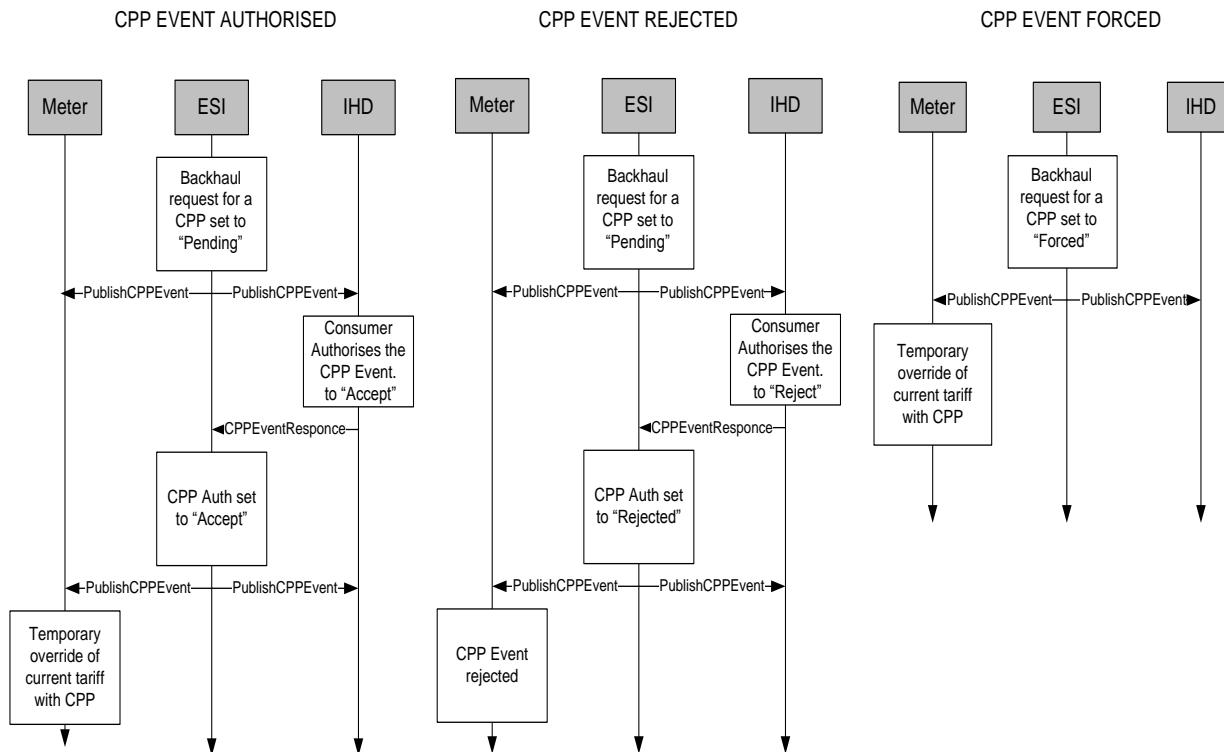


Figure D-88– CPP Event Flow

9480
 9481
 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:

1. It shall notify the consumer that there is a CPP event that requires acknowledgment. The acknowledgement shall be either to accept the *CPPEvent* or reject the *CPPEvent* (in which case it shall send the *CPPEventResponse* command, with the *CPPEventResponse* parameter set to Accepted or Rejected). It is recommended that the CPP event is ignored until a consumer either accepts or rejects the event.
2. The *CPPEventResponse* parameter is set to "Forced", in which case the *CPPEvent* has been accepted.

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.

9518 **Credit Payment Status (mandatory):** An 8-bit enumeration identifying the current credit
 9519 payment status. Refer to section D.4.2.2.9.2 for the format of this enumeration.

9520 **Credit Payment (mandatory):** An unsigned 32-bit field denoting the last credit payment. This
 9521 field should be provided in the same currency as used in the Price cluster. See also section
 9522 D.4.2.2.9.6.

9523 **Credit Payment Date (mandatory):** A UTC Time field containing the time at which the last
 9524 credit payment was made. See also section D.4.2.2.9.7.

9525 **Credit Payment Ref (mandatory):** An string of between 0-20 octets used to denote the last
 9526 credit payment reference used by the energy supplier. See also section D.4.2.2.9.8.

9527 **D.4.2.4.14 PublishCurrencyConversion Command**

9528 The *PublishCurrencyConversion* command is sent in response to a *GetCurrencyConversion*
 9529 command or when a new currency becomes available.

9530 **D.4.2.4.14.1 Payload Format**

9531 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)

9532 **Figure D-90—*PublishCurrencyConversion* Command Payload**

9533 **D.4.2.4.14.2 Payload Details**

9534 **Provider ID (mandatory):** An unsigned 32-bit field containing a unique identifier for
 9535 the commodity provider. This field allows differentiation in deregulated markets where multiple
 9536 commodity providers may be available.

9537 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. When
 9538 new information is provided that replaces older information for the same time period, this field
 9539 allows devices to determine which information is newer. The value contained in this field is a
 9540 unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)
 9541 identifying when the Publish command was issued. Thus, newer information will have a value in
 9542 the *Issuer Event ID* field that is larger than older information.

9543 **Start Time (mandatory):** A UTC Time field to denote the time at which the new currency
 9544 becomes valid. A start date/time of 0x00000000 shall indicate that the command should be
 9545 executed immediately. A start date/time of 0xFFFFFFFF shall cause an existing but pending
 9546 *PublishCurrencyConversion* command with the same *Provider ID* and *Issuer Event ID* to be
 9547 cancelled.

9548 **Old Currency (mandatory):** An unsigned 16-bit field containing identifying information
 9549 concerning the old local unit of currency used in the Price cluster. The value of the *Old Currency*
 9550 field should match the values defined by ISO 4217.

9551 **New Currency (mandatory):** An unsigned 16-bit field containing identifying information
 9552 concerning the new local unit of currency used in the Price cluster. The value of the *New*
 9553 *Currency* field should match the values defined by ISO 4217.

9554 **Conversion Factor (mandatory):** The format and use of this field is the same as for the
 9555 *ConversionFactor* attribute as defined in D.4.2.2.4.3.

9556 **Conversion Factor Trailing Digit (mandatory):** The format and use of this field is the same as
 9557 for the *ConversionFactorTrailingDigit* attribute as defined in D.4.2.2.4.4.

9558 **Currency Change Control Flags (mandatory):** A 32-bit mask that denotes the functions that
 9559 are required to be carried out on processing of this command. See Table D-114 below:

9560 **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

9561
 9562 **D.4.2.4.15 CancelTariff Command**
 9563 The *CancelTariff* command indicates that all data associated with a particular tariff instance
 9564 should be discarded.
 9565 In markets where permanently active price information is required for billing purposes, it is
 9566 recommended that replacement/superseding *PublishTariffInformation*, *PublishPriceMatrix*,

9567 *PublishBlockThresholds* and *PublishTierLabels* commands are used in place of a *CancelTariff*
 9568 command.

9569 **D.4.2.4.15.1 Payload Format**

9570 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)

9571 **Figure D-91– *CancelTariff* Command Payload**

9572 **D.4.2.4.15.2 Payload Details**

9573 **Provider ID (mandatory):** An unsigned 32-bit field containing a unique identifier for the
 9574 commodity provider. This field allows differentiation in deregulated markets where multiple
 9575 commodity providers may be available.

9576 **Issuer Tariff ID (mandatory):** Unique identifier generated by the commodity Supplier. All
 9577 parts of a tariff instance shall have the same *Issuer Tariff ID*.

9578 **Tariff Type (mandatory):** An 8-bit bitmap identifying the type of tariff to be cancelled by this
 9579 command. The least significant nibble represents an enumeration of the tariff type as detailed in
 9580 Table D-108 (Generation Meters shall use the ‘Received’ Tariff). The most significant nibble is
 9581 reserved.

9582 **D.4.2.4.15.3 Effect on Receipt**

9583 On receipt of this command, a client device shall discard all instances of
 9584 *PublishTariffInformation*, *PublishPriceMatrix*, *PublishBlockThresholds* and *PublishTierLabels*
 9585 commands associated with the stated *Provider ID*, *Tariff Type* and *Issuer Tariff ID*.

9586 **D.4.3 Client**

9587 **D.4.3.1 Dependencies**

9588 Events carried using this cluster include a timestamp with the assumption that target devices
 9589 maintain a real time clock. Devices can acquire and synchronize their internal clocks via the
 9590 ZCL Time server.

9591 If a device does not support a real time clock it is assumed that the device will interpret and
 9592 utilize the “Start Now” 0x00000000 value within the Time field.

9593 **Note:** The Price Client Cluster Attributes in this revision of this specification are provisional
 9594 and not certifiable. These features may change before reaching certifiable status in a future
 9595 revision of this specification.

9596 **D.4.3.2 Attributes**

9597 **Table D-115– Price Client Cluster Attributes**

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0000	<i>PriceIncreaseRandomizeMinutes</i>	Unsigned 8-bit Integer	0x00 to 0x3C	Read/ Write	0x05	O
0x0001	<i>PriceDecreaseRandomizeMinutes</i>	Unsigned 8-bit Integer	0x00 to 0x3C	Read/ Write	0x0F	O
0x0002	<i>Commodity Type</i>	8-bit Enumeration	0x00 to 0xFF	Read Only	-	O
0x0003 - 0x000F	Reserved					

9598

9599 **D.4.3.2.1 PriceIncreaseRandomizeMinutes Attribute**

9600 The *PriceIncreaseRandomizeMinutes* attribute represents the maximum amount of time to be
 9601 used when randomizing the response to a price increase. Note that although the granularity of
 9602 the attribute is in minutes, it is recommended the granularity of the randomization used within
 9603 a responding device be in seconds or smaller. If a device responds to a price increase it must
 9604 choose a random amount of time, in seconds or smaller, between 0 and
 9605 *PriceIncreaseRandomizeMinutes* minutes. The device must implement that random amount of
 9606 time before or after the price change. How and if a device will respond to a price increase is up
 9607 to the manufacturer. Whether to respond before or after the price increase is also up to the
 9608 manufacturer.

9609 As an example, a water heater with a *PriceIncreaseRandomizeMinutes* set to 6 could choose to
 9610 lower its set point 315 seconds (but not more than 360 seconds) before the price increases.

9611 The valid range for this attribute is 0x00 to 0x3C.

9612 If *PriceIncreaseRandomizeMinutes* or *PriceDecreaseRandomizeMinutes* attributes
 9613 are not supported by the client, then it should use the default values for the attributes as specified
 9614 in the Price Client Cluster Attribute table.

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*

9647 command without report configuration may be used for unsolicited notification of an attribute
9648 value change. Sleepy devices may have to poll.

9649 **D.4.4.3 Block Tariffs**

9650 Upon reaching the *Start Time* of a received *Publish Price* command, a device's behavior will
9651 depend on the values of the *Number of Block Thresholds* and *Number of Price Tiers* fields. A
9652 client device needing to determine if it should use Block Pricing shall send a *Get Current Price*
9653 command to the Price server and check the *Number of Block Thresholds* in the Publish Price
9654 response. Any value between 1 and 15 indicates that Block Pricing shall be used.

9655 The prices for a commodity being delivered to the premises shall be taken from the Block Pricing
9656 Information Attribute Set whenever Block Pricing is active.

9657 **D.4.4.3.1 TOU Charging Only**

9658 Indicated by the *Number of Block Thresholds* field being set to zero. Charging shall be
9659 according to the price fields within the *Publish Price* command itself.

9660 **D.4.4.3.2 Block Charging only**

9661 Indicated by the *Number of Price Tiers* fields being set to zero while the *Number of Block*
9662 *Thresholds* is between 0x01 and 0x0F.

9663 A server shall not update the Block Threshold and Block Price attribute sets of an active Block
9664 Period. Updates to these attribute sets can only be done by creating a new Block Period. The
9665 server may create a new active Block Period by updating either *Block Period Start Time*
9666 (attribute *StartOfBlockPeriod*) alone or *Block Period Duration in Minutes* (attribute
9667 *BlockPeriodDuration*) followed by *Block Period Start Time* (attribute *StartOfBlockPeriod*) along
9668 with updating other attributes as desired.

9669 When a server transmits a Publish Price command it shall additionally fill fields necessary to
9670 support backwards compatibility with clients that may not support Block Charging. The *Price*
9671 field shall be set according to the Block Price Information Attribute Set. The *Duration in Minutes*
9672 field shall be set to 0xFFFF indicating the price is valid “until changed”.

9673 A server shall additionally transmit a Publish Price command to clients under the following
9674 conditions:

- 9675 1. At the start of a Block Period
9676 2. When it is notified that a Block Threshold has been crossed
9677 3. When *Block Period Start Time* or *Block Period Duration in Minutes* have changed to
9678 indicate a new active block period

9679 A client may cache attributes from the Block Threshold, Block Period, Block Price, and Billing
9680 Period attribute sets. Cached attributes are valid only during the active Block Period when
9681 received. Upon reaching *Block Period Start Time* or detecting a new active Block Period, the
9682 client should retrieve updated values for cached attributes.

9683 A client shall check for a new active Block Period on receipt of an asynchronous Publish Price
9684 command (i.e. not required on a Publish Price command in response to Get Current Price) by
9685 checking *Block Period Start Time* and *Block Period Duration in Minutes* for update.
9686 Additionally, it shall infrequently (e.g. once an hour) query the *StartOfBlockPeriod* and
9687 *BlockPeriodDuration* attributes to verify that the Block Period has not ended early.

9688 **D.4.4.3.3 Block/TOU Combination Charging**

9689 The *Number of Block Thresholds* and *Number of Price Tiers* fields will both be set to non-zero
9690 values, indicating the number of blocks and number of tiers respectively being used. The
9691 start of a Block period shall be indicated by the value of the *Block Period Start Time* field within
9692 a *Publish Block Period* command. If the currently active parameters are not already available on the
9693 client device then, upon reaching the *Block Period Start Time*, the attributes for the required
9694 number of Block Thresholds, together with the Block Prices for all required blocks for the
9695 selected tier should be fetched from the server. The *Block Period Duration in Minutes* field
9696 shall indicate the length of the block period.

9697 A *Publish Price* command will be received for the start of each new TOU period during a block
9698 period. At this point the attributes for the Block Prices for all required blocks for the newly
9699 activated tier should be fetched from the server.

9700 Devices shall cater for both ‘blocks in tiers’ and ‘tiers in blocks’ models. In either case, the
9701 relevant prices will be defined in the Block Pricing Information Attribute Set. The ‘tiers in
9702 blocks’ model will always implement a single set of block thresholds, whereas the ‘blocks in
9703 tiers’ model may implement different thresholds for each tier.

9704 **D.4.4.3.4 Application Guidelines for Block Pricing under specific events**

9705 **HAN device not communicating with meter for extended period of time:**

9706 In this situation, when the HAN device reconnects with the meter, it will need to read the Block
9707 Information Set to calculate the correct cost for the given period. This is done by applying
9708 the prices for each block/tier combination to the consumption information for each
9709 block/tier combination. If a block period has passed while the HAN device was not
9710 communicating with the meter, then the prior period consumption information will not be
9711 known and the prior period cost cannot be calculated by the HAN device.

9712 **Meter installation or swap-out:**

9713 The new meter will need to be configured with the appropriate block thresholds, pricing, and
9714 block duration by the utility. If this does not occur precisely at the start of that customer's
9715 billing period, the utility will need to (a) pro-rate these amounts over the remaining billing
9716 period duration and (b) decide how to handle the initial portion of the period. Any information
9717 from the initial part of the billing period will be lost when the new meter is installed. As such,

9718 HAN devices may not display accurate information for this billing period and utilities should
9719 advise customers of this situation. As a typical meter lifetime is expected to be in the range of
9720 10 to 20 years, this event is expected to be rare.

9721 **D.4.4.4 Handling of Enhanced Tariffs**

9722 In ‘Traditional’ Smart Energy networks, the back-haul connection and Price server are
9723 incorporated into the meter. Fiscal accounting is out of scope of the ZigBee network. Indicative
9724 pricing information, determined by the utility or by an ESI using information supplied by the
9725 utility, is communicated from the Price server to other Smart Energy devices, using a *Publish*
9726 *Price* command, whenever the price changes.

9727 In Smart Energy networks where the meter is detached from the back-haul connection, the meter
9728 often being battery-powered and therefore unable to communicate for the majority of the time,
9729 and specifically where a Prepayment meter is required to perform independent accounting
9730 functionality, there is a need for the meter to have local access to current price and price
9731 scheduling information at all times. The optional ‘Enhanced’ tariff mechanism described in this
9732 section provides functionality to satisfy this requirement.

9733 An enhanced tariff consists of a number of commands. Depending on the mode of operation, an
9734 associated TOU calendar may also be required. *PublishBlockThresholds* and *PublishPriceMatrix*
9735 commands always include the number of block thresholds in use and number of blocks / tiers in
9736 use respectively. It is the responsibility of a client to fetch all parts belonging to a tariff after it
9737 has received an unsolicited *PublishTariffInformation* command. A client shall ensure that it
9738 successfully receives all commands associated with a tariff before any of the data for that tariff
9739 can be used. It is recommended that a client checks that the data received across all commands is
9740 valid.

9741 Whenever a new tariff is made available to a Price Server, it shall send an unsolicited
9742 *PublishTariffInformation* command to its bound clients (BOMDs shall be notified via
9743 notification flags). Other parts of the tariff (*PriceMatrix* and *BlockThresholds*) are not sent
9744 unsolicited; the clients shall send corresponding *GetPriceMatrix* and *GetBlockThresholds*
9745 commands, as applicable, to fetch the required information from the server.

9746 The Price Cluster supports different charging modes:

- 9747 • TOU charging
9748 • Block charging
9749 • TOU/Block combination charging

9750 **D.4.4.4.1 Block Charging**

9751 In case of Block charging, the following information needs to be transferred from server to
9752 client:

9753 **1** PublishTariffInformation

9754 **2** PublishPriceMatrix (noTierBlock1 .. noTierBlockN)

9755 **3** PublishBlockThresholds

9756 **4** PublishBlockPeriod

9757 In addition, a Gas-ESI may send a *PublishConversionFactor* or *PublishCalorificValue* command
9758 along with a tariff update, but does not necessarily need to.

9759 **D.4.4.4.2 TOU Charging**

9760 In case of TOU charging, the following price information needs to be transferred from server to
9761 client:

9762 **1** PublishTariffInformation

9763 **2** PublishPriceMatrix (Tier1Block1 .. TierNBlock1)

9764 **3** PublishCalendar (see D.9 for further details)

9765 In addition, a Gas-ESI may send a *PublishConversionFactor* or *PublishCalorificValue* command
9766 along with a tariff update, but does not necessarily need to.

9767 *Note: the TOU Calendar and the Tariff are linked by the start time and not by any IDs.*

9768 **D.4.4.4.3 TOU/Block Charging**

9769 In case of TOU/Block charging, the following price information needs to be transferred from
9770 server to client:

9771 **1** PublishTariffInformation

9772 **2** PublishPriceMatrix (Tier1Block1 .. TierNBlockM)

9773 **3** PublishBlockThresholds

9774 **4** PublishBlockPeriod

9775 **5** PublishCalendar (see D.9 for further details)

9776 In addition, a Gas-ESI may send a *PublishConversionFactor* or *PublishCalorificValue* command
9777 along with a tariff update, but does not necessarily need to.

9778 **D.4.4.4.4 Critical Peak Pricing**

9779 *Note: The following application guidelines that pertain to Critical Peak Pricing in this revision
9780 of this specification are provisional and not certifiable. This text may change before reaching
9781 certifiable status in a future revision of this specification.*

9782 The following additional guidelines hold for the usage of CPP events:

- 9783 • The price tiers used for CPP events (via the *PublishCPPEvent* command) are treated in the
9784 price matrix just like the ones used in the TOU Calendar. In fact, nothing prevents a tariff
9785 scheme where the same price is employed at regular times through the TOU calendar and ad-
9786 hoc via CPP events. Two prices are reserved in the price matrix for CPP events, ‘CPP1’ and
9787 ‘CPP2’
- 9788 • ESIs conforming to these specifications need to send out a *Publish Price* command along
9789 with the *PublishCPPEvent* command, for Smart Energy devices that do not support the
9790 latter.

9791 D.4.4.4.5 **Generation Charging**

9792 All Generation meters shall use the ‘Received’ sections of the Price cluster to publish the tariff
9793 information, and the Received section of the Metering Cluster.

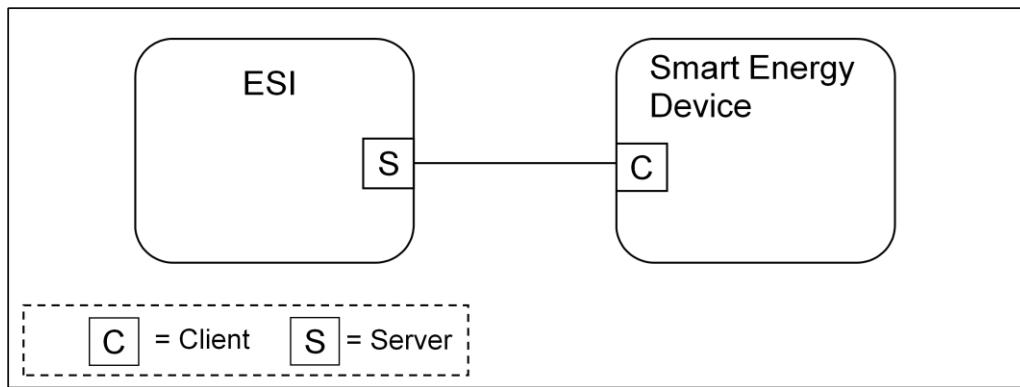
9794

9795

9796

9797 **D.5 Messaging Cluster**9798 **D.5.1 Overview**

9799 This cluster provides an interface for passing text messages between ZigBee devices.
 9800 Messages are expected to be delivered via the ESI and then unicast to all individually registered
 9801 devices implementing the Messaging Cluster on the ZigBee network, or just made available to
 9802 all devices for later pickup. Nested and overlapping messages are not allowed. The current active
 9803 message will be replaced if a new message is received by the ESI.



9804

9805 **Figure D-92– Messaging Cluster Client/Server Example**

9806 Please note the ESI is defined as the Server due to its role in acting as the proxy for upstream
 9807 message management systems and subsequent data stores.

9808 **D.5.2 Server**9809 **D.5.2.1 Dependencies**

9810 Support for ZCL Data Types.

9811 No dependencies exist for other Smart Energy Clusters.

9812 **D.5.2.2 Attributes**

9813 None.

9814 **D.5.2.3 Commands Generated**

9815 The command IDs generated by the Messaging server cluster are listed in Table D-116.

9816

9817

Table D-116– Generated Command IDs for the Messaging Server

Command Identifier Field Value	Description	Mandatory / Optional
0x00	<i>Display Message</i>	M
0x01	<i>Cancel Message</i>	M
0x02	<i>Display Protected Message</i>	O
0x03	<i>Cancel All Messages</i>	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 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 issued.

9828 **MessageControl (mandatory):** An 8-bit BitMap field indicating control information related to the 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.

	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.

- 9831
- 9832 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.
- 9833
- 9834
- 9835
- 9836 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.
- 9837
- 9838
- 9839
- 9840
- 9841
- 9842 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.
- 9843
- 9844
- 9845
- 9846
- 9847 **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.
- 9848
- 9849

9850 **Start Time (mandatory):** A UTCTime field to denote the time at which the message becomes
 9851 valid. A Start Time of 0x00000000 is a special time denoting “now.” If the device would send an
 9852 event with a Start Time of now, adjust the Duration In Minutes field to correspond to the
 9853 remainder of the event.

9854 **Duration In Minutes (mandatory):** An unsigned 16-bit field is used to denote the amount of
 9855 time in minutes after the Start Time during which the message is displayed. A Maximum value
 9856 of 0xFFFF means “until changed”.

9857 **Message (mandatory):** A ZCL String containing the message to be delivered. The String shall be
 9858 encoded in the UTF-8 format. Devices will have the ability to choose the methods for managing
 9859 messages that are larger than can be displayed (truncation, scrolling, etc.).

9860 For supporting larger messages sent over the network, both devices must agree upon a common
 9861 Fragmentation ASDU Maximum Incoming Transfer Size. Please refer to sub-clause 5.3.8 for
 9862 further details on Fragmentation settings.

9863 Any message that needs truncation shall truncate on a UTF-8 character boundary. The SE
 9864 secure payload is 59 bytes for the Message field in a non-fragmented, non-source routed
 9865 Display Message packet (11 bytes for other Display Message fields). Devices using
 9866 fragmentation can send a message larger than this. Reserving bytes for source route will reduce
 9867 this.

9868 **ExtendedMessageControl (optional):** An 8-bit BitMap field indicating additional control and
 9869 status information for a given message. Bit encoding of this field is shown in Table D-118:

9870

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		

9871

9872 The *Message Confirmation Status* bit allows the confirmation state of a message to be
 9873 communicated in the event that there are multiple IHD’s (or other Messaging cluster client
 9874 devices) on a network.

9875 The server shall initially transmit a message requiring a confirmation with the *Message*
 9876 *Confirmation Status* bit reset (0) to indicate the message had not yet been confirmed (the
 9877 *Message Confirmation* bit of the *MessageControl* field will be set to indicate a confirmation is
 9878 required).

9879 When the message is confirmed on one of the multiple IHDs in the premises, a *Message*
 9880 *Confirmation* command will be returned to the server. At this point, the server shall re-transmit

9881 the original message, but with the *Message Confirmation Status* bit now set (1) to indicate that
 9882 the message has been confirmed. This will indicate to other clients that the message no longer
 9883 requires a confirmation.

9884

9885 **D.5.2.3.2 Cancel Message Command**

9886 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)

9890 **Figure D-94—*Cancel Message* Command Payload**

9891 **D.5.2.3.2.1 Payload Details**

9892 **Message ID (mandatory):** A unique unsigned 32-bit number identifier for the message being
 9893 cancelled. It's expected the value contained in this field is a unique number managed by
 9894 upstream systems or a UTC based time stamp (UTCTime data type) identifying when the
 9895 message was originally issued.

9896 **MessageControl (mandatory):** This field is deprecated and should be set to 0x00.

9897

9898 **D.5.2.3.3 Display Protected Message Command**

9899 The *Display Protected Message* command is for use with messages that are protected by a
 9900 password or PIN.

9901 **D.5.2.3.3.1 Payload Format**

9902 The payload for this command shall be the same as that for a conventional *Display Message*
 9903 command. See D.5.2.3.1.1 for payload details.

9904

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	<i>Get Last Message</i>	M
0x01	<i>Message Confirmation</i>	M
0x02	<i>GetMessageCancellation</i>	O

0x03 – 0xff	Reserved	
-------------	----------	--

9926
99279928 D.5.3.3.1 **Get Last Message Command**

9929 This command has no payload.

9930 D.5.3.3.1.1 Effect on Receipt9931 On receipt of this command, the device shall send a *Display Message* or *Display Protected Message* command as appropriate (refer to sub-clauses D.5.2.3.1 and D.5.2.3.3). A ZCL Default 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 Utility Customer has acknowledged and/or accepted the contents of a previously sent message. Enhanced Message Confirmation commands shall contain an answer of ‘NO’, ‘YES’ and/or a message confirmation string.9940 If the optional *Message Confirmation Response* is required, the *Message Confirmation Control* 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 Details9944 **Message ID (mandatory):** A unique unsigned 32-bit number identifier for the message being confirmed.9946 **Confirmation Time (mandatory):** UTCTime of user confirmation of message.9947 **Message Confirmation Control (optional):** An 8-bit BitMap field indicating the simple confirmation that is contained within the response. Bit encoding of this field is outlined in Table D-120; if this optional field is not available, a default value of 0x00 shall be used.

9950

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– **GetMessageCancellation 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.

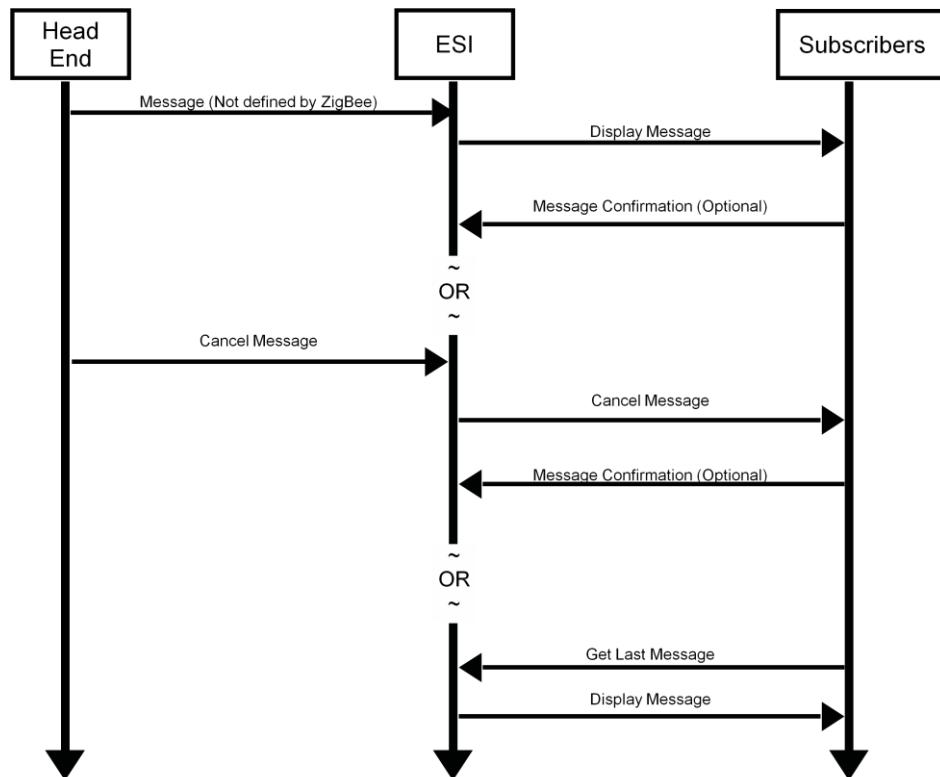
9973 A ZCL Default response with status NOT_FOUND shall be returned if there is no *Cancel All
 9974 Messages* command available that satisfies the requested criteria.

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

9983

9984 D.6 Tunneling Cluster

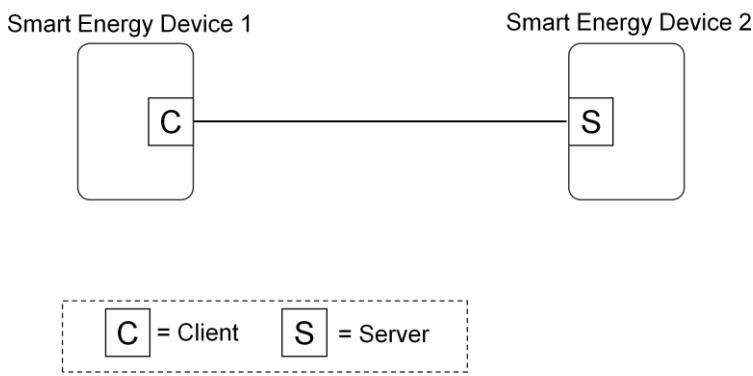
9985 **Note:** The optional support for flow control within the cluster in this revision of this
9986 specification is provisional and not certifiable. This feature set may change before reaching
9987 certifiable status in a future revision of this specification.

9988 D.6.1 Overview

9989 The tunneling cluster provides an interface for tunneling protocols. It is comprised of commands
9990 and attributes required to transport any existing metering communication protocol within
9991 the payload of standard ZigBee frames (including the handling of issues such as addressing,
9992 fragmentation and flow control). Examples for such protocols are DLMS/COSEM, IEC61107,
9993 ANSI C12, M-Bus, ClimateTalk etc.

9994 The tunneling foresees the roles of a server and a client taking part in the data exchange.
9995 Their roles are defined as follows:

- 9996 • **Client:** Requests a tunnel from the server and closes the tunnel if it is no longer needed.
- 9997 • **Server:** Provides and manages tunnels to the clients.



9998
9999
10000 **Figure D-99—A Client Requests a Tunnel From a Server to Exchange Complex
Data in Both Directions**

10001 The data exchange through the tunnel is symmetric. This means both client and server provide
10002 the commands to transfer data (*TransferData*). And both must make sure that only the partner to
10003 which the tunnel has been built up is granted read/write access to it (e.g. tunnel identifier protection
10004 through checking the MAC address).

10005 Sleepy devices either close the tunnel immediately after they have pushed their data through
10006 it, or leave it open in which case an attribute in the server (*CloseTunnelTimeout*) decides
10007 whether the tunnel is closed from the server side during the sleeping phase or not. It is
10008 recommended that battery-powered (sleepy) devices fulfil the role of the Tunneling cluster client
10009 (and therefore have control over when they request a tunnel from the server).

10010 If data is transferred to a non-existent or wrong tunnel identifier, the receiver generates an error
10011 message (*TransferDataError*).

10012 The server may support more than one tunneling protocol. The type of tunnel to be opened is a
10013 mandatory parameter (*ProtocolID*) of the tunnel request (*RequestTunnel*) that the client needs to
10014 send to the server in order to set up a new tunnel. The response from the server
10015 (*RequestTunnelResponse*) will contain a parameter with the status of the tunnel (*TunnelStatus*).
10016 If the tunnel request was successful, a unique identifier (*TunnelID*) is returned within the response.
10017 In an error case (e.g. the requested protocol is not supported) the status contains the type of
10018 error. The optional *GetSupportedTunnelProtocols* command provides a way to read out the
10019 supported protocols from the server. If the *GetSupportedTunnelProtocols* command is not
10020 supported then either the client knows the supported protocols a priori or it has to try several
10021 times using different *ProtocolIDs* until the server responds with the tunnel status *Success*.

10022 The tunneling cluster adds optional support for flow control to handle streaming protocols such
10023 as IEC61107. If implemented, flow control messages are provided to control the data flow and
10024 send acknowledges to data messages on application level. However, flow control is an optional
10025 feature and disabled per default. In the default case, the acknowledge messages
10026 (*AckTransferData*) must not be sent in order to reduce complexity and prevent from unneeded
10027 overhead.

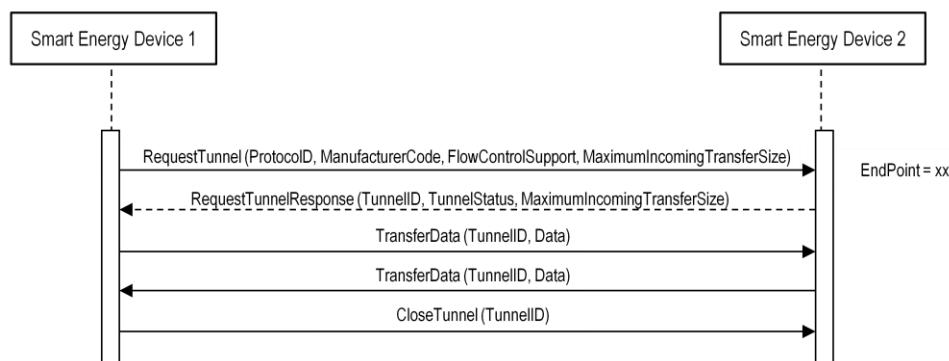
10028 The following sequence describes a typical usage:

- 10029 1 The client issues a service discovery to find devices which support the tunneling server
10030 cluster. The discovery may either be directed to one device, if its address is known, or be a
10031 broadcast (*MatchSimpleDescriptor*).
- 10032 2 The response to the discovery from the server contains an endpoint number
10033 (*SimpleDescriptor*). Using this endpoint, the client directs a tunnel request to a given server.
10034 Together with the request, the client is required to provide an enumeration with the ID of the
10035 protocol that shall be tunneled. There is the possibility to request tunnels for manufacturer
10036 specific protocols. In this case, the *ProtocolID* has to be followed by a *ZigBee*
10037 *ManufacturerCode* to open the tunnel. An additional parameter for *FlowControlSupport*
10038 accompanies the request, together with an indication of the client's incoming buffer size
10039 (*RequestTunnel* (*ProtocolID*, *ManufacturerCode*, *FlowControlSupport*,
10040 *MaximumIncomingTransferSize*)).
- 10041 3 If the server supports the protocol, it allocates the required resources, assigns a tunnel
10042 identifier and returns the ID number within the response including an additional tunnel status
10043 that the command was successful and the server's incoming buffer size. If the command
10044 failed, the status contains the reason in form of an error code (*RequestTunnelResponse*
10045 (*TunnelID*, *TunnelStatus*, *MaximumIncomingTransferSize*)). The tunnel identifier number
10046 would then be invalid in this case.

- 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

Figure D-100– SE Device 1 (Client) Requests a Tunnel From SE Device 2 (Server) to Transfer Data Without Flow Control (Default)¹⁷

¹⁷ CCB 1939



10062

10063 **Figure D-101—SE Device 1 (Client) Requests a Tunnel From SE Device 2 (Server) to Transfer Data With Flow
10064 Control¹⁸**
10065
1006610067 **D.6.2 Server**10068 **D.6.2.1 Dependencies**10069 This cluster requires APS fragmentation [B3] to be implemented, with maximum transfer sizes
10070 defined by the device's negotiated input buffer sizes.10071 **D.6.2.2 Attributes**10072 **Table D-121—Tunneling Server Attributes**

Identifier	Name	Type	Range	Access	Default	Man. /Opt.
0x0000	<i>CloseTunnelTimeout</i>	Unsigned 16-bit Integer	0x0001-0xFFFF	Read Only	0xFFFF	M

10073

¹⁸ CCB 1939

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
MaximumIncomingTransferSize	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.

10093

Table D-123– Cluster -specific Commands Received by the Server

Command Identifier FieldValue	Description	Mandatory/Optional
0x00	<i>RequestTunnel</i>	M
0x01	<i>CloseTunnel</i>	M
0x02	<i>TransferData</i>	M
0x03	<i>TransferDataError</i>	M
0x04	<i>AckTransferData</i>	O
0x05	<i>ReadyData</i>	O
0x06	<i>GetSupportedTunnelProtocols</i>	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)

Figure D-102– *RequestTunnel* Command Payload

10101

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

Table D-124– ProtocolID Enumerations

Values	Description
0	DLMS/COSEM (IEC 62056)
1	IEC 61107
2	ANSI C12
3	M-BUS
4	SML
5	ClimateTalk
6	GB-HRG
7 to 199	Reserved for future growth
200 to 254	Manufacturer-defined protocols
255	Reserved

10110
10111
10112
10113
10114
10115

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.

10116
10117
10118

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).

10119
10120

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.

10121
10122

D.6.2.4.1.3 When Generated

Is never generated by the server.

10123

D.6.2.4.1.4 Effect on Receipt

10124
10125
10126

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.

10127

D.6.2.4.2 CloseTunnel Command

10128
10129
10130
10131

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.

10132 D.6.2.4.2.1 Payload Format

Octets	2
Data Type	Unsigned 16-bit integer
Field Name	TunnelID (M)

10133 **Figure D-103– CloseTunnel Command Payload**10134 D.6.2.4.2.2 Payload Details

10135 **TunnelID:** The identifier of the tunnel that shall be closed. It is the same number that has been
 10136 previously returned in the response to a *RequestTunnel* command. Valid numbers range between
 10137 0..65535 and must correspond to a tunnel that is still active and maintained by the server.

10138 D.6.2.4.2.3 When Generated

10139 This command is never generated by the server.

10140 D.6.2.4.2.4 Effect on Receipt

10141 In case the given *TunnelID* is correct, the server closes the tunnel and frees the resources. The
 10142 associated tunnel is no longer maintained. If the *TunnelID* value does not match an active tunnel
 10143 on the server, the server shall return a ZCL Default Response with status NOT_FOUND.

10144 D.6.2.4.3 TransferData Command

10145 Command that indicates (if received) that the client has sent data to the server. The data itself
 10146 is contained within the payload.

10147 D.6.2.4.3.1 Payload Format

Octets	2	Variable
Data Type	Unsigned 16-bit integer	Octets
Field Name	TunnelID (M)	Data (M)

10148 **Figure D-104– TransferData Command Payload**10149 D.6.2.4.3.2 Payload Details

10150 **TunnelID:** A number between 0..65535 that uniquely identifies the tunnel that has been
 10151 allocated in the server triggered through the *RequestTunnel* command. This ID must be used
 10152 to send data through the tunnel or passed with any commands concerning that specific
 10153 tunnel.

10154 **Data:** Octet containing the data to be transferred through the tunnel in the format of the
 10155 communication protocol for which the tunnel has been requested and opened. The payload

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.

10184 **TransferDataStatus:** The *TransferDataStatus* parameter indicates the error that occurred
 10185 within the receiver after the last *TransferData* command.

10186 The *TransferDataStatus* values are shown in Table D-125.

10187

Table D-125– TransferDataStatus Values

Value	Description	Remarks
0x00	No such tunnel	The <i>TransferData</i> command contains a TunnelID of a non-existent tunnel.
0x01	Wrong device	The <i>TransferData</i> command contains a TunnelID that does not match the device sending the data.
0x02	Data overflow	The <i>TransferData</i> command contains more data than indicated by the <i>MaximumIncomingTransferSize</i> of the receiving device
0x03 – 0xFF	Reserved	Should not be returned and indicates an unknown error.

10188

10189 D.6.2.4.4.3 When Generated

10190 Is generated if the server wants to tell the client that there was something wrong with the last
 10191 *TransferData* command.

10192 D.6.2.4.4.4 Effect on Receipt

10193 Indicates that the client wants to tell the server that there was something wrong with the last
 10194 *TransferData* command.

10195 D.6.2.4.5AckTransferData Command

10196 Command sent in response to each *TransferData* command in case – and only in case – flow
 10197 control has been requested by the client in the *TunnelRequest* command and is supported
 10198 by both tunnel endpoints. The response payload indicates the number of octets that may still be
 10199 received by the receiver.

10200 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

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)

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

10232 for the data transfer through the tunnel or passed with any commands concerning that specific
 10233 tunnel.

10234 **NumberOfOctetsLeft:** Indicates the number of octets that may be received by the initiator of
 10235 this command (receiver). It is most likely the remaining size of the buffer holding the data that is
 10236 sent over *TransferData*. As an example: A value of 150 indicates that the next *TransferData*
 10237 command must not contain more than 150 bytes of payload or data will get lost. The value must
 10238 be larger than 0. As for its exact value, it is up to the implementer of the cluster to decide
 10239 what flow control algorithm shall be applied.

10240 D.6.2.4.6.3 When Generated

10241 If generated by the server, this command informs the client that it may now continue to send and
 10242 how much space is left within the server to receive further data.

10243 D.6.2.4.6.4 Effect on Receipt

10244 If received by the server, this command informs the server that it may now continue to send and
 10245 how much space is left within the client to receive further data.

10246 D.6.2.4.7 Get Supported Tunnel Protocols Command

10247 *Get Supported Tunnel Protocols* is the client command used to determine the tunnel protocols
 10248 supported on another device.

10249 D.6.2.4.7.1 Payload Format

Octets	1
Data Type	Unsigned 8-bit Integer
Field Name	Protocol Offset

10250 **Figure D-108—Get Supported Tunnel Protocols Command Payload**

10251 D.6.2.4.7.2 Payload Details

10252 **Protocol Offset:** Where there are more protocols supported than can be returned in a single
 10253 *Supported Tunnel Protocols Response* command, this field allows an offset to be specified on
 10254 subsequent *Get Supported Tunnel Protocols* commands. An offset of zero (0x00) should be used
 10255 for an initial (or only) *Get Supported Tunnel Protocols* command (indicating that the returned list
 10256 of protocols should commence with first available protocol). As a further example, if 10 protocols
 10257 had previously been returned, the next *Get Supported Tunnel Protocols* command should use an
 10258 offset of 10 (0x0A) to indicate the 11th available protocol should be the first returned in the
 10259 next response.

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 Response* command, indicating the tunnel protocols it supports (see sub-clause D.6.2.5.6 for 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	<i>RequestTunnelResponse</i>	M
0x01	<i>TransferData</i>	M
0x02	<i>TransferDataError</i>	M
0x03	<i>AckTransferData</i>	O
0x04	<i>ReadyData</i>	O
0x05	<i>Supported Tunnel Protocols Response</i>	O
0x06	<i>TunnelClosureNotification</i>	O

10267

10268

10269 D.6.2.5.1 RequestTunnelResponse Command

10270 *RequestTunnelResponse* is sent by the server in response to a *RequestTunnel* command previously received from the client. The response contains the status of the *RequestTunnel* command and a tunnel identifier corresponding to the tunnel that has been set-up in the server 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 allocated in the server triggered through the *RequestTunnel* command. This ID must now be

10279 used to send data through this tunnel (*TunnelID*, *TransferData*) and is also required to
 10280 close the tunnel again (*CloseTunnel*). If the command has failed, the *TunnelStatus* contains the
 10281 reason of the error and the *TunnelID* is set to 0xFFFF.

10282 **TunnelStatus:** The *TunnelStatus* parameter indicates the server's internal status after the
 10283 execution of a *RequestTunnel* command.

10284 The *TunnelStatus* values are shown in Table D-127.

10285 **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 <i>ProtocolID</i> parameter of the <i>RequestTunnel</i> command.
0x04	Flow control not supported	Flow control has been requested by the client in the <i>RequestTunnel</i> 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.

10286
 10287 **MaximumIncomingTransferSize:** A value that defines the size, in octets, of the maximum data
 10288 packet that can be transferred to the server in the payload of a single *TransferData* command.

10289 D.6.2.5.1.3 When Generated

10290 Is generated in reply to a *RequestTunnel* command to inform the client about the result of the
 10291 request.

10292 D.6.2.5.1.4 Effect on Receipt

10293 Should never be received by the server.

10294 D.6.2.5.2 TransferData Command

10295 Command that transfers data from server to the client. The data itself has to be placed within
 10296 the payload.

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)

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 TransferData Error 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

10325 supported by the device; the payload of the response should be capable of holding up to 16
 10326 protocols.

10327

10328 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

10329 **Figure D-111– Supported Tunnel Protocols Response Command Payload**

10330 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

10331 **Figure D-112– Supported Tunnel Protocols Response Command Protocol Fields**

10332 D.6.2.5.6.2 Payload Details

10333 **Protocol List Complete:** The Protocol List Complete field is a Boolean; a value of 0 indicates
 10334 that there are more supported protocols available (if more than 16 protocols are supported). A
 10335 value of 1 indicates that the list of supported protocols is complete.

10336 **Protocol Count:** The number of Protocol fields contained in the response.

10337 **Manufacturer Code:** A code that is allocated by the ZigBee Alliance, relating the manufacturer
 10338 to a device and - for tunneling - a manufacturer specific protocol. A value of 0xFFFF indicates a
 10339 standard (i.e. non- manufacturer specific) protocol

10340 **Protocol ID:** An enumeration representing the identifier of the metering communication protocol
 10341 for the supported tunnel. Table D-124 lists the possible values for standard protocols

10342 D.6.2.5.6.3 When Generated

10343 Is generated in reply to a *Get Supported Tunnel Protocols* command. to indicate the tunnel
 10344 protocols supported by the device

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.

10371

10372

10373 **D.7 Prepayment Cluster**

10374 **D.7.1 Overview**

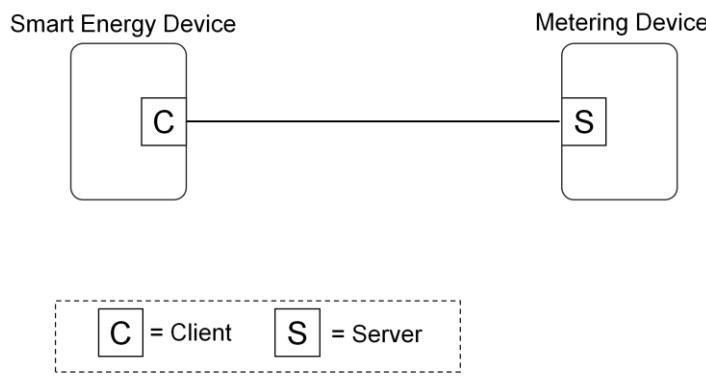
10375 The Prepayment Cluster provides the facility to pass messages relating to the accounting
 10376 functionality of a meter between devices on the HAN. It allows for the implementation of a
 10377 system conforming to the set of standards relating to Payment Electricity Meters (IEC 62055)
 10378 and 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

Figure D-114– Prepay Cluster Client Server Example

- 10401
- 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.

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.

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

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— Prepayment Information Attribute Set

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0000	<i>Payment Control Configuration</i>	16-bit Bitmap	0x0000 to 0xFFFF	Read only	0x0000	M
0x0001	<i>Credit Remaining</i>	Signed 32-bit Integer	-0x7FFFFFFF To +0x7FFFFFFF	Read only	-	O
0x0002	<i>Emergency Credit Remaining</i>	Signed 32-bit Integer	-0x7FFFFFFF to +0x7FFFFFFF	Read only	-	O
0x0003	<i>Credit Status</i>	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	0x0000000000000000 to 0xFFFFFFFFFFFFFF	Read only	-	O
0x0021	MaxCreditLimit	Unsigned 32-bit Integer	0x00000000 to 0xFFFFFFFF	Read only	-	O

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 Attribute10429 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

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 supply on expiry of available credit.

10436

10437 **Prepayment Enabled:** Indicates if the meter is a 'prepayment' meter; if this value is 0, the meter is considered to be a 'credit' meter.

- 10439 **Credit Management Enabled:** Indicates whether the metering device should manage
10440 accounting functionality according to available tariff information.
- 10441 **Credit Display Enabled:** Indicates whether the metering device should display the credit
10442 status.
- 10443 **Account Base:** Indicates whether the metering device is running in Monetary (0) or Unit based
10444 (1) units. If Monetary based, the unit of measure is defined in the Price cluster, if Unit based,
10445 the unit of measure is defined in the Metering cluster
- 10446 **Contactor Fitted:** Indicates whether the metering device is fitted with a Contactor i.e. is
10447 capable if disconnecting the energy supply.
- 10448 **Standing Charge Configuration:** Indicates whether the standing charge collection is halted
10449 when the prepaid credit is exhausted.
- 10450 **Emergency Standing Charge Configuration:** Indicates whether the standing charge collection
10451 is halted when the device is in Emergency Credit mode.
- 10452 **Debt Configuration:** Indicates whether the debt collection is halted when the prepaid credit is
10453 exhausted.
- 10454 **Emergency Debt Configuration:** Indicates whether the debt is collected when the device is in
10455 Emergency Credit mode.
- 10456 D.7.2.2.1.2 Credit Remaining Attribute
10457 The *Credit Remaining* attribute represents the amount of credit remaining on the Metering
10458 Device. If Monetary-based, this attribute is measured in a base unit of *Currency* with the
10459 decimal point located as indicated by the *Trailing Digits* field, as defined in the Price cluster. If
10460 Unit-based, the unit of measure is as defined in the Metering cluster (see sub-clause D.3.2.2.4.1).
- 10461 D.7.2.2.1.3 Emergency Credit Remaining Attribute
10462 The *Emergency Credit Remaining* attribute represents the amount of Emergency Credit still
10463 available on the Metering Device. If Monetary-based, this attribute is measured in a base unit of
10464 *Currency* with the decimal point located as indicated by the *Trailing Digits* field, as defined in
10465 the Price cluster. If Unit-based, the unit of measure is as defined in the Metering cluster (see sub-
10466 clause D.3.2.2.4.1).
- 10467 D.7.2.2.1.4 Credit Status Attribute
10468 The *Credit Status* attribute represents the current status of credit within the Metering Device.
10469 Bit encoding of this field is outlined in Table D-131. Explanation of the use of this attribute can
10470 be found in section D.7.4.1.

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

Saturday	£5	Cap reached	£5
Sunday	£5	Cap reached	£5

10491
10492 Once the cap value has been reached during a week then no further amounts are deducted from
10493 the purchases.

10494 As an extension of the example, if the customer purchased £50 credit on the Monday, the meter
10495 would take £5 (not £10) and would also not take any further debt payments from any other
10496 purchases made in the same week.

10497 D.7.2.2.1.8 EmergencyCreditLimit/Allowance Attribute

10498 The *EmergencyCreditLimit/Allowance* attribute may be updated by the utility company. This is
10499 the amount of Emergency Credit available to loan to the consumer when the remaining balance
10500 goes below the low credit threshold. If Monetary based, then this attribute is measured in base
10501 unit of *Currency* with the decimal point located as indicated by the *Trailing Digits* field, as
10502 defined in the Price cluster. If Unit based, the unit of measure is as defined in the Metering
10503 cluster (see sub-clause D.3.2.2.4.1).

10504 D.7.2.2.1.9 EmergencyCreditThreshold Attribute

10505 When credit (or emergency credit) falls below this threshold, an alarm is raised to warn the
10506 consumer of imminent supply interruption and, if available, to offer Emergency Credit. If
10507 Monetary based, the unit of measure is the same as that defined in the Price cluster. If Unit based,
10508 the unit of measure is as defined in the Metering cluster (see sub-clause D.3.2.2.4.1).

10509 D.7.2.2.1.10 TotalCreditAdded Attribute

10510 An unsigned 48-bit integer value indicating running total of credit topped up to date. If Monetary
10511 based, this attribute is measured in a base unit of *Currency* with the decimal point located as
10512 indicated by the *Trailing Digits* field, as defined in the Price cluster. If Unit based, the unit of
10513 measure is as defined in the Metering cluster (see sub-clause D.3.2.2.4.1). At change of Tenant or
10514 Supplier, this attribute shall be reset to zero.

10515 D.7.2.2.1.11 MaxCreditLimit Attribute

10516 An unsigned 32-bit integer value indicating the maximum credit balance allowed on a meter.
10517 Any further top-up amount that will cause the meter to exceed this limit will be rejected. This
10518 value can be stated in currency (as per the Price cluster) or in units (unit of measure will be
10519 defined in the Metering cluster) depending on the Prepayment mode of operation defined in
10520 section D.7.2.2.1.1 (*Payment Control Configuration* attribute). A value of 0xFFFFFFFF shall
10521 indicate that this limit is disabled and that all further top-ups should be permitted.

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).

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 cases). This allowance 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).

10564 This attribute represents a threshold relating to the absolute value of the *CreditRemaining* attribute, that when reached (when credit is decrementing) causes the supply of service to be disconnected. There can be several types of credit within a payment metering system of which there are 2 specified in this specification (*Credit* and *EmergencyCredit*). The *CreditRemaining* attribute shall contain the net worth of a consumers account within the meter, consolidating all active credit types (both *Credit* and *EmergencyCredit* if in use). As *EmergencyCredit* is effectively a loan from the supplier it becomes a liability once it is used, and when it is exhausted will force the *RemainingCredit* to a negative value. There are a number of other factors that can affect the way a prepayment meter works and which values are displayed to the end consumer. However, when a meter's *EmergencyCredit* has run out, the *CreditRemaining* value shall contain the total liability of the consumer (that he is required to pay before *EmergencyCredit* shall be 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 that is used with this meter set. The *TokenCarrierId* attribute is a ZCL Octet String capable of storing a 20 character string (the first Octet indicates length) encoded in the UTF-8 format. The *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 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	<i>Top up Date/Time #1</i>	UTCTime		Read only	-	O

0x0101	<i>Top up Amount #1</i>	Signed 32-bit Integer	-0x7FFFFFFF to +0x7FFFFFFF	Read only	-	O
0x0102	<i>Originating Device #1</i>	8 bits Enumeration	0x00 to 0xFF	Read only	-	O
0x0103	<i>Top up Code #1</i>	Octet String	1-26	Read only	-	O
0x0104-0x010F	Reserved					
0x0110	<i>Top up Date/Time #2</i>	UTCTime		Read only	-	O
0x0111	<i>Top up Amount #2</i>	Signed 32-bit Integer	-0x7FFFFFFF to +0x7FFFFFFF	Read only	-	O
0x0112	<i>Originating Device #2</i>	8 bits Enumeration	0x00 to 0xFF	Read only	-	O
0x0113	<i>Top up Code #2</i>	Octet String	1-26	Read only	-	O
0x0114-0x011F	Reserved					
0x0120	<i>Top up Date/Time #3</i>	UTCTime		Read only	-	O
0x0121	<i>Top up Amount #3</i>	Signed 32-bit Integer	-0x7FFFFFFF to +0x7FFFFFFF	Read only	-	O
0x0122	<i>Originating Device #3</i>	8 bits Enumeration	0x00 to 0xFF	Read only	-	O
0x0123	<i>Top up Code #3</i>	Octet String	1-26	Read only	-	O
0x0124-0x012F	Reserved					
0x0130	<i>Top up Date/Time #4</i>	UTCTime		Read only	-	O
0x0131	<i>Top up Amount #4</i>	Signed 32-bit Integer	-0x7FFFFFFFFFFFF to +0x7FFFFFFFFFFFF	Read only	-	O

0x0132	<i>Originating Device #4</i>	8 bits Enumeration	0x00 to 0xFF	Read only	-	O
0x0133	<i>Top up Code #4</i>	Octet String	1-26	Read only	-	O
0x0134-0x013F	Reserved					
0x0140	<i>Top up Date/Time#5</i>	UTCTime		Read only	-	O
0x0141	<i>Top up Amount #5</i>	Signed 32-bit Integer	-0x7FFFFFFF to +0x7FFFFFFF	Read only	-	O
0x0142	<i>Originating Device #5</i>	8 bits Enumeration	0x00 to 0xFF	Read only	-	O
0x0143	<i>Top up Code #5</i>	Octet String	1-26	Read only	-	O
0x0144-0x01FF	Reserved					

10585

10586 D.7.2.2.1 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 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.7.2.2.3 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.

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

0x0223	DebtRecoveryStartTime#2	UTCTime		Read only	-	O
0x0224	DebtRecoveryCollectionTime#2	Unsigned 16-bit integer	0x0000 – 0x05A0	Read only	0	O
0x0225	Reserved					
0x0226	DebtRecoveryFrequency#2	8 bits Enumeration	0x00 – 0xFF	Read only	-	O
0x0227	DebtRecoveryAmount#2	Unsigned 32-bit Integer	0x00000000 – 0xFFFFFFFF	Read only	-	O
0x0228	Reserved					
0x0229	DebtRecoveryTopUpPercentage#2	Unsigned 16-bit integer	0x0000 – 0x2710	Read only	0	O
0x022A – 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	DebtRecoveryMethod#3	8 bits Enumeration	0x00 – 0xFF	Read only	-	O
0x0233	DebtRecoveryStartTime#3	UTCTime		Read only	-	O
0x0234	DebtRecoveryCollectionTime#3	Unsigned 16-bit integer	0x0000 – 0x05A0	Read only	0	O
0x0235	Reserved					
0x0236	DebtRecoveryFrequency#3	8 bits Enumeration	0x00 – 0xFF	Read only	-	O
0x0237	DebtRecoveryAmount#3	Unsigned 32-bit Integer	0x00000000 – 0xFFFFFFFF	Read only	-	O
0x0238	Reserved					
0x0239	DebtRecoveryTopUpPercentage#3	Unsigned 16-bit integer	0x0000 – 0x2710	Read only	0	O
0x023A – 0x02FF	Reserved					

10609

10610 D.7.2.2.3.1 DebtLabel#N Attribute10611 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

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.

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.

Table D-136–Recovery Frequency Field Enumerations

Enumerated Value	Recovery Period
0x00	Per Hour
0x01	Per Day
0x02	Per Week

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 0xFFFF	Read/Write	0xFFFF	O
0x0403	PrepayEventAlarmMask	16-bit BitMap	0x0000 to 0xFFFF	Read/Write	0xFFFF	O

10660

10661 D.7.2.2.5.1 Prepayment Alarm Status Attribute

10662 The *PrepaymentAlarmStatus* attribute provides indicators reflecting the current error conditions found by the prepayment metering device. This attribute is a 16-bit field where when an individual bit is set, an error or warning condition exists. The behaviour causing the setting or resetting of each bit is device specific. In other words, the application within the prepayment metering device will determine and control when these settings are either set or cleared. The ESI should make alarms available to upstream systems, together with consumption data collected 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

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)

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

0x0500	HistoricalCostConsumptionFormatting	8-bit BitMap	0x00 to 0xFF	Read Only	-	O
0x0501	ConsumptionUnitofMeasurement	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 0xFFFF	Read Only	-	O
0x0504-0x051B	Reserved					
0x051C	CurrentDayCostConsumptionDelivered	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x051D	CurrentDayCostConsumptionReceived	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x051E	PreviousDayCostConsumptionDelivered	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x051F	PreviousDayCostConsumptionReceived	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0520	PreviousDay2CostConsumptionDelivered	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0521	PreviousDay2CostConsumptionReceived	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0522	PreviousDay3CostConsumptionDelivered	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0523	PreviousDay3CostConsumptionReceived	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0524	PreviousDay4CostConsumptionDelivered	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0525	PreviousDay4CostConsumptionReceived	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O

0x0526	PreviousDay5 CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFFF	Read Only	-	O
0x0527	PreviousDay5 CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFFF	Read Only	-	O
0x0528	PreviousDay6 CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFFF	Read Only	-	O
0x0529	PreviousDay6 CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFFF	Read Only	-	O
0x052A	PreviousDay7 CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFFF	Read Only	-	O
0x052B	PreviousDay7 CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFFF	Read Only	-	O
0x052C	PreviousDay8 CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFFF	Read Only	-	O
0x052D	PreviousDay8 CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFFF	Read Only	-	O
0x052E- 0x052F	Reserved					
0x0530	CurrentWeek CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFFF	Read Only	-	O
0x0531	CurrentWeek CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFFF	Read Only	-	O
0x0532	PreviousWeek CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFFF	Read Only	-	O
0x0533	PreviousWeek CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFFF	Read Only	-	O
0x0534	PreviousWeek2 CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFFF	Read Only	-	O
0x0535	PreviousWeek2 CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFFF	Read Only	-	O

0x0536	PreviousWeek3 CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFFF	Read Only	-	O
0x0537	PreviousWeek3 CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFFF	Read Only	-	O
0x0538	PreviousWeek4 CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFFF	Read Only	-	O
0x0539	PreviousWeek4 CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFFF	Read Only	-	O
0x053A	PreviousWeek5 CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFFF	Read Only	-	O
0x053B	PreviousWeek5 CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFFF	Read Only	-	O
0x053C- 0x053F	Reserved					
0x0540	CurrentMonth CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFFF	Read Only	-	O
0x0541	CurrentMonth CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFFF	Read Only	-	O
0x0542	PreviousMonth CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFFF	Read Only	-	O
0x0543	PreviousMonth CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFFF	Read Only	-	O
0x0544	PreviousMonth2 CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFFF	Read Only	-	O
0x0545	PreviousMonth2 CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFFF	Read Only	-	O
0x0546	PreviousMonth3 CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFFF	Read Only	-	O

0x0547	PreviousMonth3 CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFFF	Read Only	-	O
0x0548	PreviousMonth4 CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFFF	Read Only	-	O
0x0549	PreviousMonth4 CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFFF	Read Only	-	O
0x054A	PreviousMonth5 CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFFF	Read Only	-	O
0x054B	PreviousMonth5 CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFFF	Read Only	-	O
0x054C	PreviousMonth6 CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFFF	Read Only	-	O
0x054D	PreviousMonth6 CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFFF	Read Only	-	O
0x054E	PreviousMonth7 CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFFF	Read Only	-	O
0x054F	PreviousMonth7 CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFFF	Read Only	-	O
0x0550	PreviousMonth8 CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFFF	Read Only	-	O
0x0551	PreviousMonth8 CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFFF	Read Only	-	O
0x0552	PreviousMonth9 CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFFF	Read Only	-	O
0x0553	PreviousMonth9 CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFFF	Read Only	-	O
0x0554	PreviousMonth10 CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFFF	Read Only	-	O

0x0555	PreviousMonth10 CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFFF	Read Only	-	O
0x0556	PreviousMonth11 CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFFF	Read Only	-	O
0x0557	PreviousMonth11 CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFFF	Read Only	-	O
0x0558	PreviousMonth12 CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFFF	Read Only	-	O
0x0559	PreviousMonth12 CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFFF	Read Only	-	O
0x055A	PreviousMonth13 CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFFF	Read Only	-	O
0x055B	PreviousMonth13 CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFFF	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*

- 10706 • *PreviousWeekNCostConsumptionDelivered*
10707 • *PreviousWeekNCostConsumptionReceived*
10708 • *CurrentMonthCostConsumptionDelivered*
10709 • *CurrentMonthCostConsumptionReceived*
10710 • *PreviousMonthNCostConsumptionDelivered*
10711 • *PreviousMonthNCostConsumptionReceived*

10712
10713 D.7.2.2.6.2 ConsumptionUnitofMeasurement Attribute

10714 *ConsumptionUnitofMeasurement* provides a label for the Energy, Gas, or Water being measured
10715 by the metering device. This attribute is an 8-bit enumerated field. The bit descriptions for this
10716 attribute are listed in Table D-26.

10717 This attribute shall be used against the following attributes:

- 10718 • *CurrentDayCostConsumptionDelivered*
10719 • *CurrentDayCostConsumptionReceived*
10720 • *PreviousDayNCostConsumptionDelivered*
10721 • *PreviousDayNCostConsumptionReceived*
10722 • *CurrentWeekCostConsumptionDelivered*
10723 • *CurrentWeekCostConsumptionReceived*
10724 • *PreviousWeekNCostConsumptionDelivered*
10725 • *PreviousWeekNCostConsumptionReceived*
10726 • *CurrentMonthCostConsumptionDelivered*
10727 • *CurrentMonthCostConsumptionReceived*
10728 • *PreviousMonthNCostConsumptionDelivered*
10729 • *PreviousMonthNCostConsumptionReceived*

10730
10731 D.7.2.2.6.3 CurrencyScalingFactor Attribute

10732 *CurrencyScalingFactor* provides a scaling factor for the *Currency* attribute for the Energy, Gas,
10733 or Water being measured by the metering device. This attribute is an 8-bit enumeration, the
10734 enumerated values for which are outlined in Table D-144. Note that this attribute will allow for a
10735 different resolution for historical values compared to values in the Price cluster.

10736 This attribute shall be used against the following attributes:

- 10737 • *CurrentDayCostConsumptionDelivered*
10738 • *CurrentDayCostConsumptionReceived*
10739 • *PreviousDayNCostConsumptionDelivered*
10740 • *PreviousDayNCostConsumptionReceived*

- 10741 • *CurrentWeekCostConsumptionDelivered*
 10742 • *CurrentWeekCostConsumptionReceived*
 10743 • *PreviousWeekNCostConsumptionDelivered*
 10744 • *PreviousWeekNCostConsumptionReceived*
 10745 • *CurrentMonthCostConsumptionDelivered*
 10746 • *CurrentMonthCostConsumptionReceived*
 10747 • *PreviousMonthNCostConsumptionDelivered*
 10748 • *PreviousMonthNCostConsumptionReceived*

10749
10750

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

10751

10752

10753 **D.7.2.2.6.4 Currency Attribute**

10754 The *Currency* attribute provides the currency for the Energy, Gas, or Water being measured by
 10755 the prepayment device. The value of the attribute should match one of the values defined by ISO
 10756 4217. This unsigned 16-bit value indicates the currency in which the following attributes are
 10757 represented:

- 10758 • *CurrentDayCostConsumptionDelivered*
 10759 • *CurrentDayCostConsumptionReceived*
 10760 • *PreviousDayNCostConsumptionDelivered*
 10761 • *PreviousDayNCostConsumptionReceived*
 10762 • *CurrentWeekCostConsumptionDelivered*

-
- 10763 • *CurrentWeekCostConsumptionReceived*
 10764 • *PreviousWeekNCostConsumptionDelivered*
 10765 • *PreviousWeekNCostConsumptionReceived*
 10766 • *CurrentMonthCostConsumptionDelivered*
 10767 • *CurrentMonthCostConsumptionReceived*
 10768 • *PreviousMonthNCostConsumptionDelivered*
 10769 • *PreviousMonthNCostConsumptionReceived*
 10770

10771 D.7.2.2.6.5 CurrentDayCostConsumptionDelivered Attribute

10772 *CurrentDayCostConsumptionDelivered* represents the summed value of Energy, Gas, or Water
 10773 delivered to the premises since the HFT. If optionally provided,
 10774 *CurrentDayCostConsumptionDelivered* is updated continuously as new measurements are made.
 10775 If the optional *Historical Freeze Time* attribute is not available, default to midnight local time.

10776 D.7.2.2.6.6 CurrentDayCostConsumptionReceived Attribute

10777 *CurrentDayCostConsumptionReceived* represents the summed value of Energy, Gas, or Water
 10778 received from the premises since the HFT. If optionally provided,
 10779 *CurrentDayCostConsumptionReceived* is updated continuously as new measurements are made.
 10780 If the optional *Historical Freeze Time* attribute is not available, default to midnight local time.

10781 D.7.2.2.6.7 PreviousDayNCostConsumptionDelivered Attribute

10782 *PreviousDayNCostConsumptionDelivered* represents the summed value of Energy, Gas, or Water
 10783 Water delivered to the premises within the previous 24 hour period starting at the HFT. If the
 10784 optional *Historical Freeze Time* attribute is not available, default to midnight local time.

10785 D.7.2.2.6.8 PreviousDayNCostConsumptionReceived Attribute

10786 *PreviousDayNCostConsumptionReceived* represents the summed value of Energy, Gas, or Water
 10787 received from the premises within the previous 24 hour period starting at the HFT. If the optional
 10788 *Historical Freeze Time* attribute is not available, default to midnight local time.

10789 D.7.2.2.6.9 CurrentWeekCostConsumptionDelivered Attribute

10790 *CurrentWeekCostConsumptionDelivered* represents the summed value of Energy, Gas, or Water
 10791 delivered to the premises since the HFT on Monday to the last HFT read. If optionally provided,
 10792 *CurrentWeekCostConsumptionDelivered* is updated continuously as new measurements are
 10793 made. If the optional *Historical Freeze Time* attribute is not available, default to midnight local
 10794 time.

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 optionally
10820 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.

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	<i>Select Available Emergency Credit</i>	O
0x01	Reserved	
0x02	<i>Change Debt</i>	O
0x03	<i>Emergency Credit Setup</i>	O
0x04	<i>Consumer Top Up</i>	O
0x05	<i>CreditAdjustment</i>	O
0x06	<i>Change Payment Mode</i>	O
0x07	<i>Get Prepay Snapshot</i>	O
0x08	<i>Get Top Up Log</i>	O
0x09	<i>Set Low Credit Warning Level</i>	O
0x0A	<i>Get Debt Repayment Log</i>	O
0x0B	<i>Set Maximum Credit Limit</i>	O
0x0C	<i>Set Overall Debt Cap</i>	O

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).

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)

Figure D-116– Change Debt Command Payload

10876 D.7.2.3.3.2 **Payload Details**

10877 **Issuer Event Id (mandatory):** Unique identifier generated by the commodity provider. When
10878 new information is provided that replaces older information for the same time period, this field
10879 allows devices to determine which information is newer. The value contained in this field is a
10880 unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)
10881 identifying when the command was issued. Thus, newer information will have a value in the
10882 *Issuer Event ID* field that is larger than older information.

10883 **DebtLabel (mandatory):** The format and use of this field is the same as for the *DebtLabel#N*
10884 attribute as defined in D.7.2.2.3.1. A value of 0xFF in the first Octet (length) shall indicate that
10885 the value of this parameter shall remain unchanged on the Metering device following receipt of
10886 this command.

10887 **DebtAmount (mandatory):** The format and use of this field is the same as for the
10888 *DebtAmount#N* attribute as defined in D.7.2.2.3.2. A *DebtAmount* of 0xFFFFFFFF shall indicate
10889 that the value of this parameter shall remain unchanged on the Metering device following receipt
10890 of this command.

10892 **DebtRecoveryMethod (mandatory):** The format and use of this field is the same as for the
 10893 *DebtRecoveryMethod#N* attribute as defined in D.7.2.2.3.3. A *DebtRecoveryMethod* of 0xFF
 10894 shall indicate that the value of this parameter shall remain unchanged on the Metering device
 10895 following receipt of this command.

10896 **DebtAmountType (mandatory):** An 8-bit enumeration field identifying the type of debt
 10897 information to be issued within this command. The Types are detailed in Table D-147 below:

10898 **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

10899 **DebtRecoveryStartTime (mandatory):** The format and use of this field is the same as for the
 10900 *DebtRecoveryStartTime#N* attribute as defined in D.7.2.2.3.4. A *DebtRecoveryStartTime* of
 10901 0xFFFFFFFF shall indicate that the value of this parameter shall remain unchanged on the
 10902 Metering device following receipt of this command.

10904 **DebtRecoveryCollectionTime (mandatory):** The format and use of this field is the same as for
 10905 the *DebtRecoveryCollectionTime#N* attribute as defined in D.7.2.2.3.5. A *DebtRecoveryCollectionTime* of
 10906 0xFFFF shall indicate that the value of this parameter shall remain unchanged on the Metering
 10907 device following receipt of this command.

10908 **DebtRecoveryFrequency (mandatory):** The format and use of this field is the same as for the
 10909 *DebtRecoveryFrequency#N* attribute as defined in D.7.2.2.3.6. A *DebtRecoveryFrequency* of
 10910 0xFF shall indicate that the value of this parameter shall remain unchanged on the Metering
 10911 device following receipt of this command. Note that the value of this field is unused when the
 10912 *DebtRecoveryMethod* is set to *Percentage Based*.

10913 **DebtRecoveryAmount (mandatory):** The format and use of this field is the same as for the
 10914 *DebtRecoveryAmount#N* attribute as defined in D.7.2.2.3.7. A *DebtRecoveryAmount* of
 10915 0xFFFFFFFF shall indicate that the value of this parameter shall remain unchanged on the
 10916 Metering device following receipt of this command.

10917 **DebtRecoveryBalancePercentage (mandatory):** The format and use of this field is the same as
 10918 for the *DebtRecoveryTopUpPercentage#N* attribute as defined in D.7.2.2.3.8. A

10919 *DebtRecoveryBalancePercentage* of 0xFFFF shall indicate that the value of this parameter shall
 10920 remain unchanged on the Metering device following receipt of this command.

10921 D.7.2.3.3.3 When Generated

10922 This command is generated when there is a change to the debt, which the Head End System
 10923 requires to be sent down to the meter.

10924

10925 D.7.2.3.4 Emergency Credit Setup Command

10926 This command provides a method to set up the parameters for the Emergency Credit.

10927 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)

10928 **Figure D-117– Emergency Credit Setup Command Payload**

10929 D.7.2.3.4.2 Payload Details

10930 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. When
 10931 new information is provided that replaces older information for the same time period, this field
 10932 allows devices to determine which information is newer. The value contained in this field is a
 10933 unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)
 10934 identifying when the command was issued. Thus, newer information will have a value in the
 10935 *Issuer Event ID* field that is larger than older information.

10936 **Start Time (mandatory):** A UTC Time field to denote the time at which the Emergency Credit
 10937 settings become valid. A start date/time of 0x00000000 shall indicate that the command should
 10938 be executed immediately. A start date/time of 0xFFFFFFFF shall cause an existing but pending
 10939 *Emergency Credit Setup* command with the same *Issuer Event ID* to be cancelled.

10940 **Emergency Credit Limit (allowance) (mandatory):** An unsigned 32-bit field to denote the
 10941 Emergency Credit limit on the Metering Device, measured in base unit of *Currency* (as per the
 10942 Price cluster) or in Units (as per the Metering cluster) with the decimal point located as indicated
 10943 by the *TrailingDigits* field, as defined in the Price cluster. When no Emergency Credit has been
 10944 used, this is the value defined within the *EmergencyCreditRemaining* attribute (D.7.2.2.1.3).

10945 **Emergency Credit Threshold (mandatory):** An unsigned 32-bit field to denote the amount of
 10946 credit remaining on the Metering Device below which the Emergency Credit facility can be
 10947 selected. The value is measured in base unit of *Currency* (as per the Price cluster) or in Units (as

10948 per the Metering cluster) with the decimal point located as indicated by the *TrailingDigits* field,
 10949 as defined in the Price cluster.

10950 D.7.2.3.4.3 When Generated

10951 The *Emergency Credit Setup* command is used when the Head End System has a requirement to
 10952 change the Prepayment configuration on the meter.

10953

10954 D.7.2.3.5 Consumer Top Up Command

10955 The *Consumer Top Up* command is used by the IHD and the ESI as a method to apply credit top
 10956 up values to a prepayment meter.

10957 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)

10958 **Figure D-118– Consumer Top Up Command Payload**

10959 D.7.2.3.5.2 Payload Details

10960 **Originating Device (mandatory):** An 8 bit enumeration field identifying the Smart Energy
 10961 device issuing the selection command, as defined in Table D-146.

10962 **Top Up Code (mandatory):** An octet string of between 1 and 26 characters (the first character
 10963 indicates the string length).

10964 D.7.2.3.5.3 When Generated

10965 The *Consumer Top Up* command shall be generated when a new Top-up amount of credit has
 10966 been purchased from the energy supplier and is required to be sent to the Meter. Alternatively,
 10967 the command can be used to transfer an instruction such as to connect or disconnect the supply,
 10968 enable a particular display sequence, or other action via an appropriate *Top Up (UTRN) Code*.

10969 D.7.2.3.5.4 Effect on Receipt

10970 The meter shall update the *Top Up Date/Time#1*, *Top Up Amount#1* and the *Originating*
 10971 *Device#1* attributes on the valid processing of this command. It shall then send the
 10972 *ConsumerTopUpResponse* command to all devices bound to the cluster.

10973 A Mirroring device receiving this command shall return a ZCL Default Response with a status
 10974 code of NOTIFICATION_PENDING. If the command buffer on the mirror is already full, the
 10975 mirror shall instead return a ZCL Default Response to the initiating device with a status code of
 10976 INSUFFICIENT_SPACE. The Mirroring device may timeout the buffered message, in which

10977 case it shall return a ZCL Default Response with a status code of TIMEOUT (see D.3.4.4.3 for
10978 further details).

10979

10980 D.7.2.3.6 **Credit Adjustment Command**

10981 The *Credit Adjustment* command is sent to update the *Credit Remaining* attribute on a
10982 Prepayment meter. It shall only be sent from an ESI to the Meter.

10983 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)

10984 **Figure D-119– Credit Adjustment Command Payload**

10985 D.7.2.3.6.2 **Payload Details**

10986 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. When
10987 new information is provided that replaces older information for the same time period, this field
10988 allows devices to determine which information is newer. The value contained in this field is a
10989 unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)
10990 identifying when the command was issued. Thus, newer information will have a value in the
10991 *Issuer Event ID* field that is larger than older information.

10992 **Start Time (mandatory):** A UTC Time field to denote the time at which the credit adjustment
10993 settings become valid. A start date/time of 0x00000000 shall indicate that the command should
10994 be executed immediately. A start date/time of 0xFFFFFFFF shall cause an existing but pending
10995 *Credit Adjustment* command with the same *Issuer Event ID* to be cancelled.

10996 **Credit Adjustment Type (mandatory):** An 8-bit enumeration field identifying the type of
10997 credit adjustment to be issued out within this command. The Types are detailed within Table
10998 D-148 below.

10999 **Table D-148– Credit Type Field Enumerations**

Enumerated Value	Credit Type
0x00	Credit Incremental
0x01	Credit Absolute
0x02 – 0xFF	Reserved

11000

11001 **Credit Adjustment Value (mandatory):** A signed 32-bit field to denote the value of the credit
 11002 adjustment, measured in base unit of *Currency* (as per the Price cluster) or in Units (as per the
 11003 Metering cluster) with the decimal point located as indicated by the *TrailingDigits* field, as
 11004 defined in the Price cluster. This can be a positive or negative value.

11005 D.7.2.3.6.3 When Generated

11006 The *Credit Adjustment* command shall be sent to the meter when the ESI has a new credit
 11007 adjustment value for the meter.

11008 D.7.2.3.6.4 Effect on Receipt

11009 The *Credit Adjustment Value* shall be used to update the *Credit Remaining* attribute to the correct
 11010 value.

11011

11012 D.7.2.3.7 Change Payment Mode Command

11013 This command is sent to a Metering Device to instruct it to change its mode of operation, e.g.
 11014 from Credit to Prepayment.

11015 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)

11016 Figure D-120– Change Payment Mode Command Payload

11017 D.7.2.3.7.2 Payload Details

11018 **Provider ID (mandatory):** An unsigned 32 bit field containing a unique identifier for the
 11019 commodity supplier to whom this command relates.

11020 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. When
 11021 new information is provided that replaces older information for the same time period, this field
 11022 allows devices to determine which information is newer. The value contained in this field is a
 11023 unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)
 11024 identifying when the command was issued. Thus, newer information will have a value in the
 11025 *Issuer Event ID* field that is larger than older information.

11026 **Implementation Date/Time (mandatory):** A UTC Time field to indicate the date from which
11027 the payment mode change is to be applied. An *Implementation Date/Time* value of 0x00000000
11028 shall indicate that the command should be executed immediately. An *Implementation Date/Time*
11029 value of 0xFFFFFFFF shall cause an existing but pending *Change Payment Mode* command with
11030 the same *Provider ID* and *Issuer Event ID* to be cancelled.

11031 **Proposed Payment Control Configuration (mandatory):** An 16-bit BitMap indicating the
11032 actions required in relation to switching the payment mode. Bit encoding of this field is outlined
11033 in Table D-130.

11034 **Cut off Value (mandatory):** The format and use of this field is the same as for the *CutOffValue*
11035 attribute as defined in D.7.2.2.1.19. A *CutOffValue* of 0xFFFFFFFF shall indicate that the value
11036 of this parameter shall remain unchanged on the Metering device following receipt of this
11037 command.

11038 D.7.2.3.7.3 When Generated

11039 The *Change Payment Mode* command shall be sent from the Energy Supplier, via the ESI, only
11040 when the need to change the mode of the meter arises.

11041 D.7.2.3.7.4 Effect on Receipt

11042 On receipt of the *ChangePaymentMode* command, the meter shall send the
11043 *ChangePaymentModeResponse*. The meter should create all snapshots required before the mode
11044 is changed and transmit these to the ESI. It should then also create all required snapshots and
11045 request valid Price, TOU and Prepayment information (refer to sections D.3.2.3.1.7 and D.7.2.4.2
11046 for further details).

11047

11048 D.7.2.3.8 Get Prepay Snapshot Command

11049 This command is used to request the cluster server for snapshot data.

11050 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

11052 D.7.2.3.8.2 Payload Details

11053 **Earliest Start Time (mandatory):** A UTC Timestamp indicating the earliest time of a snapshot
11054 to be returned by a corresponding *Publish Prepay Snapshot* command. Snapshots with a time
11055 stamp equal to or greater than the specified *Earliest Start Time* shall be returned.

11056 **Latest End Time (mandatory):** A UTC Timestamp indicating the latest time of a snapshot to be
 11057 returned by a corresponding *Publish Prepay Snapshot* command. Snapshots with a time stamp
 11058 less than the specified *Latest End Time* shall be returned.

11059 **Snapshot Offset (mandatory):** Where multiple snapshots satisfy the selection criteria specified
 11060 by the other fields in this command, this field identifies the individual snapshot to be returned.
 11061 An offset of zero (0x00) indicates that the first snapshot satisfying the selection criteria should be
 11062 returned, 0x01 the second, and so on.

11063 **Snapshot Cause (mandatory):** This field is used to request only snapshots for a specific cause.
 11064 The allowable values are listed in Table D-151. Setting the type to 0xFFFFFFFF indicates that all
 11065 snapshots should be transmitted, irrespective of the cause.

11066 D.7.2.3.8.3 [Effect on Receipt](#)

11067 On receipt of this command, the server will respond with the appropriate data as detailed in sub-
 11068 clause D.7.2.4.2.

11069 A ZCL Default Response with status NOT_FOUND shall be returned if the server does not have
 11070 a snapshot which satisfies the received parameters (e.g. no snapshot with a timestamp between
 11071 the *Earliest Start Time* and the *Latest End Time*).

11072

11073 D.7.2.3.9 [Get Top Up Log](#)

11074 This command is sent to the Metering Device to retrieve the log of Top Up codes received by the
 11075 meter.

11076 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)

11077 **Figure D-122– Get Top Up Code Log Command Payload**

11078 D.7.2.3.9.2 [Payload Details](#)

11079 **Latest End Time (mandatory):** UTC timestamp indicating the latest *TopUp Time* of Top
 11080 Up records to be returned by the corresponding *Publish Top Up Log* commands. The
 11081 first returned Top Up record shall be the most recent record with its *TopUp Time* equal
 11082 to or older than the *Latest End Time* provided.

11083 **Number of Records (mandatory):** An 8-bit integer which represents the maximum number of
 11084 records that the client is willing to receive in response to this command. A value of 0 would

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 the value 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**

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

Field Name	Provider ID (M)	Issuer Event ID (M)	Implementation Date/Time (M)	Maximum Credit Level (M)	Maximum Credit Per Top Up (M)
------------	-----------------	---------------------	------------------------------	--------------------------	-------------------------------

Figure D-125– Set Maximum Credit Level Command Payload

11134

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.

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	<i>Reserved</i>	O

0x01	<i>Publish Prepay Snapshot</i>	O
0x02	<i>Change Payment Mode Response</i>	O
0x03	<i>Consumer Top Up Response</i>	O
0x04	<i>Reserved</i>	O
0x05	<i>Publish Top Up Log</i>	O
0x06	<i>Publish Debt Log</i>	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.

11210 **Command Index (mandatory):** The *Command Index* is used 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.

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.

11221 **SnapshotPayloadType (mandatory):** The *SnapshotPayloadType* is an 8-bit enumerator
 11222 defining the format of the *SnapshotPayload* in this message. The different snapshot types are
 11223 listed in Table D-152. The server selects the *SnapshotPayloadType* based on the charging
 11224 scheme in use.

11225 **Table D-152– Snapshot Payload Type**

Enumeration	Description
0x00	Debt/Credit Status
0x01 – 0xFE	Reserved
0xFF	Not used

11226

11227 **SnapshotPayload (mandatory):** the format of the *SnapshotPayload* differs depending on the
 11228 *SnapshotPayloadType*.

11229 **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)

11230 **Figure D-128– Debt/Credit Status Command Payload**

11231 **Accumulated Debt (mandatory):** The *AccumulatedDebt* field represents the total amount of
 11232 debt remaining on the Metering Device, measured in a base unit of *Currency* with the decimal
 11233 point located as indicated by the *Trailing Digits* field, as defined in the Price cluster.

11234 **Type 1 Debt Remaining (mandatory):** The *Type1DebtRemaining* field represents the amount of
 11235 Type 1 debt remaining on the Metering Device, measured in base unit of *Currency* with the
 11236 decimal point located as indicated by the *TrailingDigits* field, as defined in the Price cluster.

11237 **Type 2 Debt Remaining (mandatory):** The *Type2DebtRemaining* field represents the amount of
 11238 Type 2 debt remaining on the Metering Device, measured in base unit of *Currency* with the
 11239 decimal point located as indicated by the *TrailingDigits* field, as defined in the Price cluster.

11240 **Type 3 Debt Remaining (mandatory):** The *Type3DebtRemaining* field represents the amount of
 11241 Type 3 debt remaining on the Metering Device, measured in base unit of *Currency* with the
 11242 decimal point located as indicated by the *TrailingDigits* field, as defined in the Price cluster.

11243 **Emergency Credit Remaining (mandatory):** The *EmergencyCreditRemaining* field represents
 11244 the amount of Emergency Credit still available on the Metering Device. If Monetary based, this
 11245 field is measured in a base unit of *Currency* (as per the Price cluster) or in Units (as per the

11246 Metering cluster), with the decimal point located as indicated by the *TrailingDigits* field, as
 11247 defined in the Price cluster. If Unit based, the unit of measure is as defined in the Metering
 11248 cluster (see sub-clause D.3.2.2.4.1).

11249 **Credit Remaining (mandatory):** The *CreditRemaining* field represents the amount of credit
 11250 remaining on the Metering Device. If Monetary based, this field is measured in a base unit of
 11251 Currency (as per the Price cluster) or in Units (as per the Metering cluster), with the decimal
 11252 point located as indicated by the *TrailingDigits* field, as defined in the Price cluster. If Unit
 11253 based, the unit of measure is as defined in the Metering cluster (see sub-clause D.3.2.2.4.1).

11254

11255 **D.7.2.4.3 Change Payment Mode Response Command**

11256 This command is sent in response to the *ChangePaymentMode* command. The
 11257 *ChangePaymentModeResponse* command shall only inform the ESI of the current default setting
 11258 that would affect the meter when entering into Prepayment/PAYG or Credit mode. Should these
 11259 values require changing then other commands within the Prepayment & Price cluster should be
 11260 used.

11261 **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)

11262 **Figure D-129– Change Payment Mode Response Command Payload**

11263 **D.7.2.4.3.2 Payload Details**

11264 **Friendly Credit (mandatory):** An 8-bit BitMap to show if the meter has a Friendly Credit
 11265 calendar and that this calendar shall be enabled.

11266

Table D-153– Friendly Credit BitMap

Bit	Description
0	Friendly credit enabled
1-7	Reserved

11267

11268 **Friendly Credit Calendar ID (mandatory):** An unsigned 32-bit field to denote the
 11269 *IssuerCalendarID* that shall be used for the friendly credit periods. The *IssuerCalendarID* can be
 11270 found within the TOU cluster (see Annex D.9).

11271 **Emergency Credit Limit/Allowance (mandatory):** An unsigned 32-bit field to denote the
 11272 emergency credit limit on the Metering Device, measured in base unit of *Currency* with the
 11273 decimal point located as indicated by the *TrailingDigits* field, as defined in the Price cluster.
 11274 Should no emergency credit have been used, this is the value defined within the
 11275 *EmergencyCreditRemaining* attribute (D.7.2.2.1.3).

11276 **Emergency Credit Threshold (mandatory):** An unsigned 32-bit field to denote the amount of
 11277 credit remaining on the Metering Device below which the *Emergency Credit* facility can be
 11278 selected. The value is measured in base unit of *Currency* with the decimal point located as
 11279 indicated by the *TrailingDigits* field, as defined in the Price cluster.

11280 D.7.2.4.3.3 When Generated

11281 The *ChangePaymentModeResponse* command is generated in response to a
 11282 *ChangePaymentMode* command.

11283

11284 D.7.2.4.4 Consumer Top Up Response Command

11285 The Metering device responds either with the following values in the case of a credit token
 11286 received:

- 11287 • Meter's enumerated status, after receiving the top up, in the *Result Type* field
- 11288 • Received Top up token's credit value in the *Top Up Value* field
- 11289 • The source of the top up, enumerated in the *Source of Top up* field
- 11290 • The credit remaining on the meter, after the addition of this Top Up, in the *Credit Remaining*
 11291 field,

11292 OR, in the case of a connect/disconnect Top Up (UTRN) code, with the following:

- 11293 • Supply status, after processing of the token, enumerated in the *Result Type* field
- 11294 • Top up token's credit value SET TO ZERO in the *Top Up Value* field
- 11295 • The source of the top up, enumerated in the *Source of Top up* field
- 11296 • The credit remaining on the meter, after the addition of this Top Up, in the *Credit Remaining*
 11297 field

11298 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)

11299 Figure D-130– Consumer Top Up Response Command Payload

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.

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 processed¹⁹. 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

- 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.

11367 **Amount Collected (mandatory):** An unsigned 32-bit field to denote the amount of debt
11368 collected at this time. This parameter shall be measured in base unit of *Currency* with the
11369 decimal point located as indicated by the *Trailing Digits* field, as defined in the Price cluster.

11370 **Debt Type (mandatory):** An 8-bit enumeration field identifying the type of debt the record
11371 refers to. The enumerations are defined in Table D-149.

11372 **Outstanding Debt (mandatory):** An unsigned 32-bit field to denote the amount of debt still
11373 outstanding after the debt was collected. This parameter shall be measured in base unit of
11374 *Currency* with the decimal point located as indicated by the *Trailing Digits* field, as defined in
11375 the Price cluster.

11376

11377 **D.7.3 Client**

11378 **D.7.3.1 Dependencies**

- 11379 • Support for ZCL Data Types
- 11380 • Events carried using this cluster include a timestamp with the assumption that target devices
11381 maintain a real time clock. Devices can acquire and synchronize their internal clocks via the
11382 ZCL Time server.

11383 **D.7.3.2 Attributes**

11384 The client has no attributes.

11385 **D.7.3.3 Commands Received**

11386 The client receives the cluster-specific response commands detailed in D.7.2.4.

11387 **D.7.3.4 Commands Generated**

11388 The client generates the cluster-specific commands detailed in D.7.2.3, as required by the
11389 application.

11390

11391 **D.7.4 Application Guidelines**

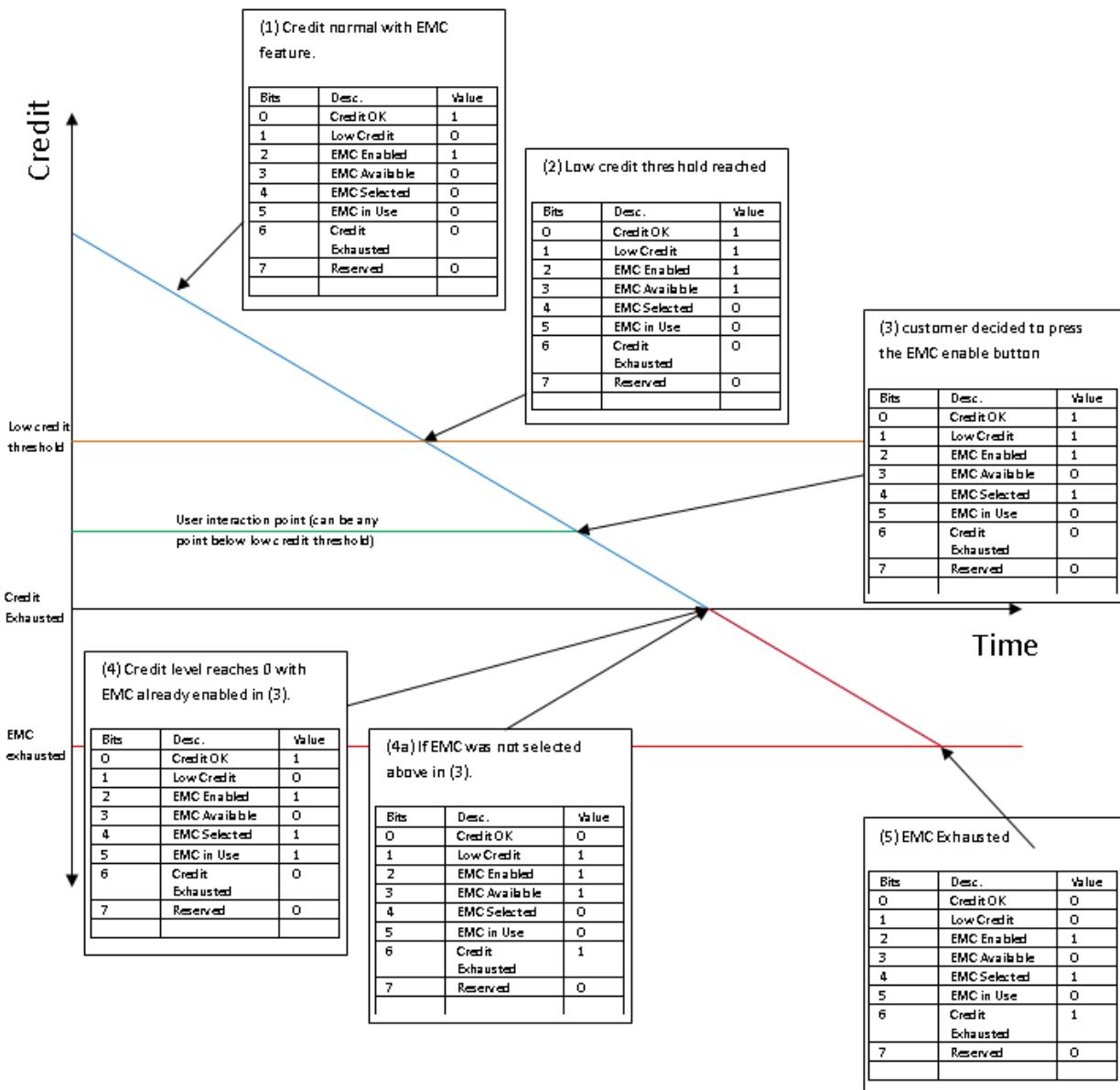
11392 **D.7.4.1 Credit Status Attribute**

11393 The purpose of the *Credit Status* attribute is to describe to any device on the HAN, what the
11394 status of a meter operating in Prepayment mode may be at any point in time. There are a number
11395 of important functionalities in Prepayment meters, and a variety of implementations depending
11396 on the manufacturer and their chosen system, however this attribute is designed to pick up the

11397 lowest common denominator of statuses that would be important to an end user looking to glean
11398 information about their meter in the HAN. For example, has their meter run out of credit, is
11399 Emergency Credit available or has Emergency Credit been selected?

11400 The diagram below describes the manner in which this attribute SHOULD be used when
11401 describing these statuses and others. This guidance note is not designed to prescribe how any
11402 Prepayment meter logic works, but merely to get a common understanding of the meter status to
11403 the end users' interface device. It is entirely up to device manufacturers to decide how to best use
11404 this information and display it.

11405



11406
11407

Figure D-135– Prepayment Credit Status Attribute Explained

11408

11409 D.7.4.1.1 Statuses Explained - an Example

11410 Below is a brief explanation of each status noted on the diagram above in order to give a better
 11411 indication of what the meter is doing at any given point. Imagine that the diagonal blue line
 11412 represents the customer's credit, and when it turns into a plum colored diagonal line below the
 11413 Time-Axis, the reader can assume that the meter is in negative credit, and Emergency Credit may
 11414 or may not be invoked depending on the use case.

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).

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

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**

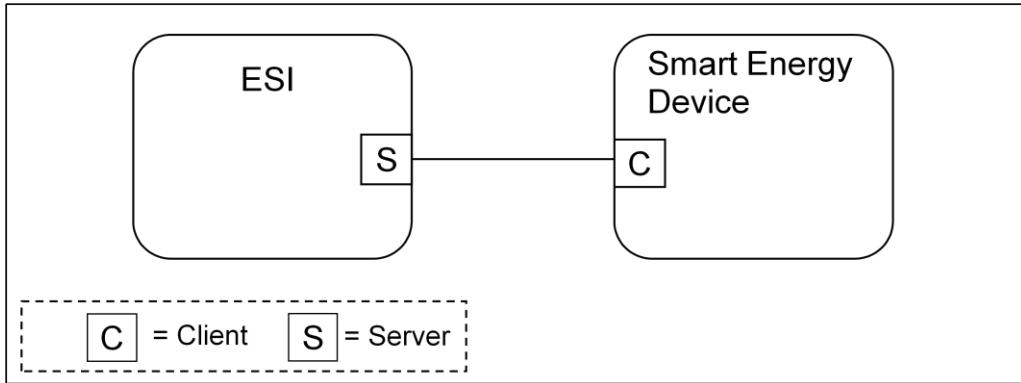
11481 The OTA cluster defines the message formatting used to pass device images but does not
11482 specify when to use the cluster. The following policies specify how and when to use the OTA
11483 cluster such that all devices in an SE network will upgrade at predictable intervals.

- 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.

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.

11521 The Calendar cluster will support all of the following calendar types:

11522 • Delivered

11523 • Received

11524 • Delivered and Received

11525 • Friendly Credit

11526 • Auxiliary Load Switch

11527 Each calendar has three associated tables, a Season table, a Week Profile table and a Day Profile
 11528 table. These are described in Table D-155. In addition, there is a Special Day Table which allows
 11529 special days to be defined (days where a special switching behavior overrides the normal
 11530 operation). Each entry in the Special Day table contains a date together with the Day ID for a
 11531 Day Profile (in the associated Calendar's Day Profile table) to be used on that date.

11532 **Table D-155—Calendar Data Structures**

Table	Description
Season Table	<p>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.</p> <p>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.</p> <p>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.</p>
Week Profile Table	<p>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.</p> <p>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.</p>
Day Profile Table	<p>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.</p>
Special Day Table	<p>Defines special dates. On such dates, a special switching behavior overrides the normal one defined by the Season and Week Profile Tables.</p> <p>The Day Profile referenced through the Day ID in the Special Days Table activates the Day Schedule of the corresponding Day Profile.</p>

11533

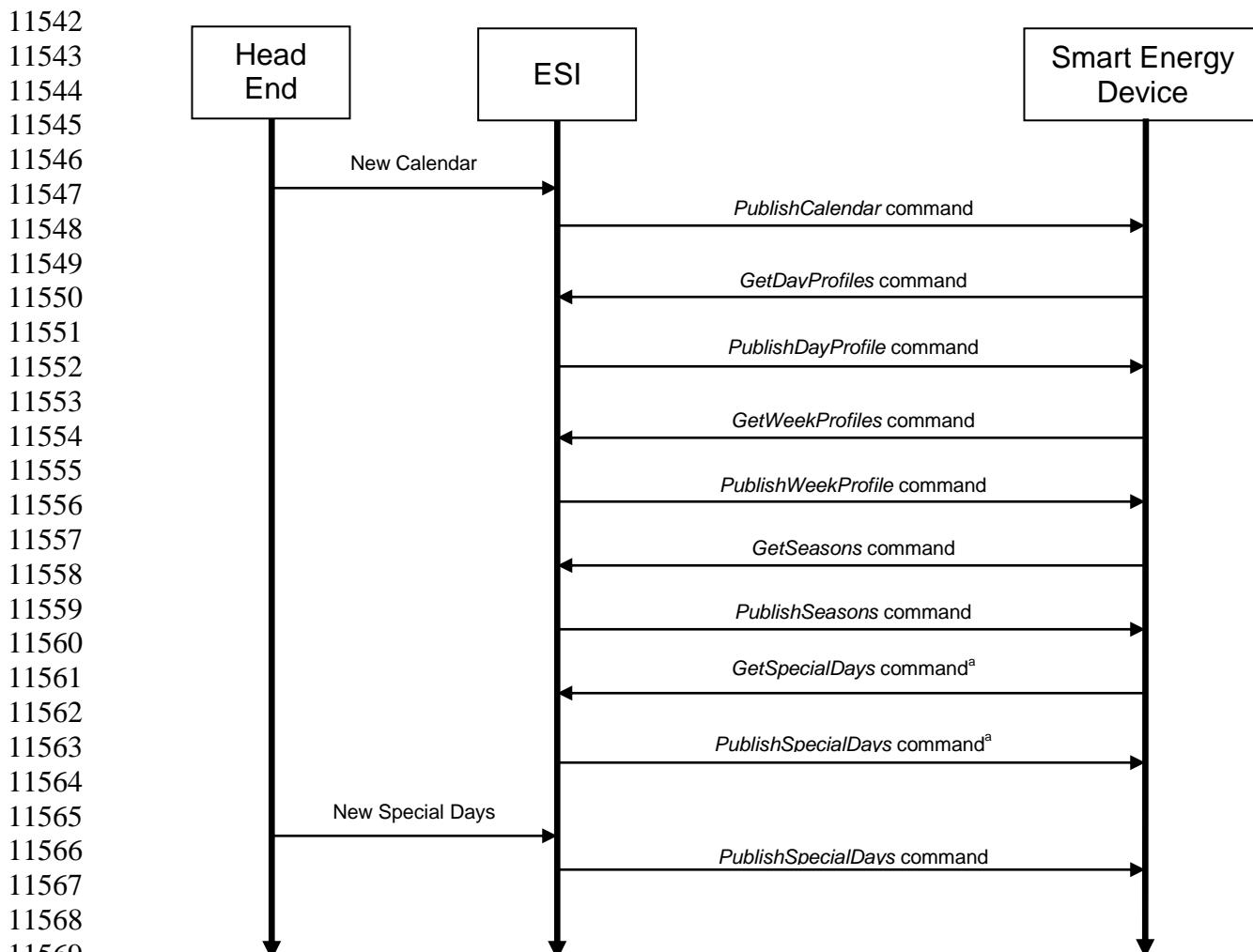
11534

11535 All dates and times shall be defined according to UTC, Standard or Local time. Alternatively, the
11536 Season Table may be used to accommodate requirements such as daylight saving.

11537
11538 Figure D-137 shows a recommended Calendar command sequence (noting that this sequence is
11539 for a main-powered Smart Energy Device) :

11540

11541



^aAlthough not necessary, it is thought wise to check for updates when a new calendar is published

Figure D-137– Recommended Calendar Command Sequence

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

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 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.

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 Payload11617 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.

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

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)

Octets	1	Variable
Data Type	8-bit Enumeration	Series of Schedule Entries

Field Name	Calendar Type (M)	Day Schedule Entries
------------	-------------------	----------------------

11677

Figure D-139 – PublishDayProfile Command Payload

11678

D.9.2.3.2.2 Payload Details

11679
11680
11681

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.

11682
11683
11684
11685
11686
11687

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.

11688
11689

Issuer Calendar ID (mandatory): Unique identifier generated by the commodity supplier. All parts of a calendar instance shall have the same *Issuer Calendar ID*.

11690
11691
11692

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.

11693
11694

Total Number of Schedule Entries (mandatory): An 8-bit integer representing the total number of *ScheduleEntries* in this Day Profile.

11695
11696
11697

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.

11698
11699
11700

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.

11701
11702
11703

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.

11704
11705
11706

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

11707 Friendly Credit Start Times shall be used. If the value is 0x04 then the Auxilliary Load Start
 11708 Times shall be used. A value other than these would be invalid.

11709 **D.9.2.3.2.3.1 Schedule Entries for Rate Start Times**

11710 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)

11711 **Figure D-140– Schedule Entries for Rate Start Times Command Sub-Payload**

11712 **Start Time (mandatory):** The *Start Time* is represented in minutes from midnight.
 11713 *ScheduleEntries* must be arranged in ascending order of *Start Times*. The first Schedule Entry
 11714 must have 0x0000 (midnight) as the StartTime.

11715 **Price Tier (mandatory):** This is the current price tier that is valid until the start time of the next
 11716 Schedule Entry.

11717 **D.9.2.3.2.3.2 Schedule Entries for Friendly Credit Start Times**

11718 A *Friendly Credit Start Time* entry consists of a start time and an indication if Friendly Credit is
 11719 available.

Octets	2	1
Data Type	Unsigned 16-bit Integer	Boolean
Field Name	Start Time (M)	Friendly Credit Enable (M)

11720 **Figure D-141– Schedule Entries for Friendly Credit Start Times Command Sub-Payload**

11721 **Start Time (mandatory):** The *Start Time* is represented in minutes from midnight.
 11722 *ScheduleEntries* must be arranged in ascending order of *Start Times*. The first Schedule Entry
 11723 must have 0x0000 (midnight) as the StartTime.

11724 **Friendly Credit Enable (mandatory):** The *Friendly Credit Enable* field is a Boolean denoting
 11725 if the Friendly Credit period is available for the consumer to use. A value of 1 means it is enabled
 11726 and a 0 means that the Friendly Credit period is not available for the consumer to use.

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)

1						
----------	----------	----------	----------	----------	----------	----------

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 Payload11751 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:

-
- 11780 • The *SeasonEntries* shall be arranged in a linear array ordered by the date.
- 11781 • The first command shall have *Command Index* set to 0, the second to 1 and so on.
- 11782 • The total number of seasons being transferred with the whole set of commands is known
- 11783 from the previously received *PublishCalendar* command.
- 11784 • All associated commands shall use the same value of *Issuer Event ID*.

11785 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

11786 **Figure D-144—*PublishSeasons* Command Payload**

11787 D.9.2.3.4.2 [Payload Details](#)

11788 **Provider Id (mandatory):** An unsigned 32-bit field containing a unique identifier for the

11789 commodity provider. This field allows differentiation in deregulated markets where multiple

11790 commodity providers may be available.

11791 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. When

11792 new information is provided that replaces older information for the same time period, this field

11793 allows devices to determine which information is newer. The value contained in this field is a

11794 unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)

11795 identifying when the Publish command was issued. Thus, newer information will have a value in

11796 the *Issuer Event ID* field that is larger than older information.

11797 **Issuer Calendar ID (mandatory):** Unique identifier generated by the commodity supplier. All

11798 parts of a calendar instance shall have the same *Issuer Calendar ID*.

11799 **Command Index (mandatory):** The *Command Index* is used to count the payload fragments in

11800 the case where the entire payload does not fit into one message. The *Command Index* starts at 0

11801 and is incremented for each fragment belonging to the same command.

11802 **Total Number of Commands (mandatory):** In the case where the entire payload does not fit

11803 into one message, the *Total Number of Commands* field indicates the total number of sub-

11804 commands in the message.

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)

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.

If the maximum application payload is not sufficient to transfer all *SpecialDayEntries* in one command, the ESI may send as many *PublishSpecialDays* commands as needed. In this case:

- The *SpecialDayEntries* 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 SpecialDays* field shall be set in all commands to the total number of entries being transferred with the whole set of commands.
 - 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

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.

11856 **Command Index (mandatory):** The *Command Index* is used to count the payload fragments in
11857 the case where the entire payload does not fit into one message. The *Command Index* starts at 0
11858 and is incremented for each fragment belonging to the same command.

11859 **Total Number of Commands (mandatory):** In the case where the entire payload does not fit
11860 into one message, the *Total Number of Commands* field indicates the total number of sub-
11861 commands in the message.

11862 **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
11863 ascending order.
11864

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

11866

11867 D.9.2.3.6 CancelCalendar Command

11868 The *CancelCalendar* command indicates that all data associated with a particular calendar
11869 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.

11875 D.9.2.3.6.1 Payload Format

11876 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

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
--------	---	---	---	---	---

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

- 11901
- 11902 D.9.2.4.1.2 Payload Details
- 11903 **Earliest Start Time (mandatory):** UTC Timestamp indicating the earliest start time of
11904 calendars to be returned by the corresponding *PublishCalendar* command. The first returned
11905 *PublishCalendar* command shall be the instance which is active or becomes active at or after the
11906 stated *Earliest Start Time*. If more than one instance is requested, the active and scheduled
11907 instances shall be sent with ascending ordered *Start Time*.
- 11908 **Min. Issuer Event ID (mandatory):** A 32-bit integer representing the minimum *Issuer Event ID*
11909 of calendars to be returned by the corresponding *PublishCalendar* command. A value of
11910 0xFFFFFFFF means not specified; the server shall return calendars irrespective of the value of
11911 the *Issuer Event ID*.
- 11912 **Number of Calendars (mandatory):** An 8-bit integer which represents the maximum number of
11913 *PublishCalendar* commands that the client is willing to receive in response to this command. A
11914 value of 0 would indicate all available *PublishCalendar* commands shall be returned.
- 11915 **Calendar Type (mandatory):** An 8-bit enumeration identifying the calendar type of the
11916 requested calendar. Generation Meters shall use the ‘Received’ Calendar. See Table D-159. A
11917 value of 0xFF means not specified. If the *CalendarType* is not specified, the server shall return
11918 calendars regardless of its type.
- 11919 **Provider Id (mandatory):** An unsigned 32-bit field containing a unique identifier for the
11920 commodity provider. This field allows differentiation in deregulated markets where multiple
11921 commodity providers may be available. A value of 0xFFFFFFFF means not specified; the server
11922 shall return calendars irrespective of the value of the *Provider Id*.

11923 D.9.2.4.2 GetDayProfiles Command

11924 This command initiates one or more *PublishDayProfile* commands for the referenced Calendar.

11925 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

Field Name	Provider Id (M)	Issuer Calendar ID (M)	Start Day Id (M)	Number of Days (M)
------------	-----------------	------------------------	------------------	--------------------

11926

Figure D-150– GetDayProfiles Command Payload

11927 D.9.2.4.2.2 Payload Details

11928 **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*.

11932 **Issuer Calendar ID (mandatory):** *IssuerCalendarID* of the calendar to which the requested Day Profiles belong.

11934 **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.

11939 **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*.

11942 Note: A Day Profile table may need multiple *PublishDayProfile* commands to be transmitted to the client.

11944 D.9.2.4.3 GetWeekProfiles Command

11945 This command initiates one or more *PublishWeekProfile* commands for the referenced Calendar.

11946 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)

11948

Figure D-151– GetWeekProfiles Command Payload

11949 D.9.2.4.3.2 Payload Details

11950 **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

11952 commodity providers may be available. A value of 0xFFFFFFFF means not specified; the server
 11953 shall return week profiles irrespective of the value of the *Provider Id*.

11954 **Issuer Calendar ID (mandatory):** *IssuerCalendarID* of the calendar to which the requested
 11955 Week Profiles belong.

11956 **Start Week ID (mandatory):** Unique identifier for a Week Profile generated by the commodity
 11957 supplier. The *Start Week ID* indicates the minimum ID of Week Profiles to be returned by the
 11958 corresponding *PublishWeekProfile* command. A value of 0x01 indicates that the
 11959 *PublishWeekProfile* command should contain the profile with the lowest Week ID held by the
 11960 server. A value of 0x00 is unused.

11961 **Number of Weeks (mandatory):** An 8-bit integer which represents the maximum number of
 11962 Week Profiles that the client is willing to receive in response to this command. A value of 0x00
 11963 will cause the return of all week profiles with an ID equal to or greater than the *Start Week ID*.

11964 **D.9.2.4.4 GetSeasons Command**

11965 This command initiates one or more *PublishSeasons* commands for the referenced Calendar.

11966 **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)

11967 **Figure D-152– GetSeasons Command Payload**

11968 **Provider Id (mandatory):** An unsigned 32-bit field containing a unique identifier for the
 11969 commodity provider. This field allows differentiation in deregulated markets where multiple
 11970 commodity providers may be available. A value of 0xFFFFFFFF means not specified; the server
 11971 shall return season tables irrespective of the value of the *Provider Id*.

11972 **Issuer Calendar ID (mandatory):** *IssuerCalendarID* of the calendar to which the requested
 11973 Seasons belong.

11974 Note: A Season Table may need multiple *PublishSeasons* commands to be transmitted to the
 11975 client.

11976 **D.9.2.4.5 GetSpecialDays Command**

11977 This command initiates one or more *PublishSpecialDays* commands for the scheduled Special
 11978 Day Table updates.

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)

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.

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.

12034 Battery-powered devices, or device with limited resources, should not bind to the Calendar
12035 cluster. These devices are expected to poll the Calendar server regularly in order to check for
12036 updates to calendar items.

12037 It is recommended that calendar information is persisted on devices throughout a reboot or
12038 power-cycle. However, ALL devices should request the latest calendar information following
12039 power up, after a reboot, or following any period without HAN communication.

12040 Acquisition of a calendar starts when a client asks for or gets pushed a current or pending
12041 *PublishCalendar* command. From the information contained in the *PublishCalendar* command,
12042 the client should request the relevant day, week and/or season information, respectively utilizing
12043 *GetDayProfiles*, *GetWeekProfiles* and *GetSeasons* commands.

12044 There may be specific days when special switching behavior overrides the normal one defined by
12045 the Season or Week Profile tables. These special dates are contained within a Special Day Table
12046 associated with the particular calendar instance. As Special Day Table information may change
12047 more frequently than the other information contained within a calendar, any update to the Special
12048 Day Table will be sent unsolicited to Calendar clients registered with the relevant Calendar
12049 server. Battery-powered devices are expected to poll the Calendar server regularly for updates to
12050 the Special Day information in a similar way to that used for other calendar information.

12051

12052

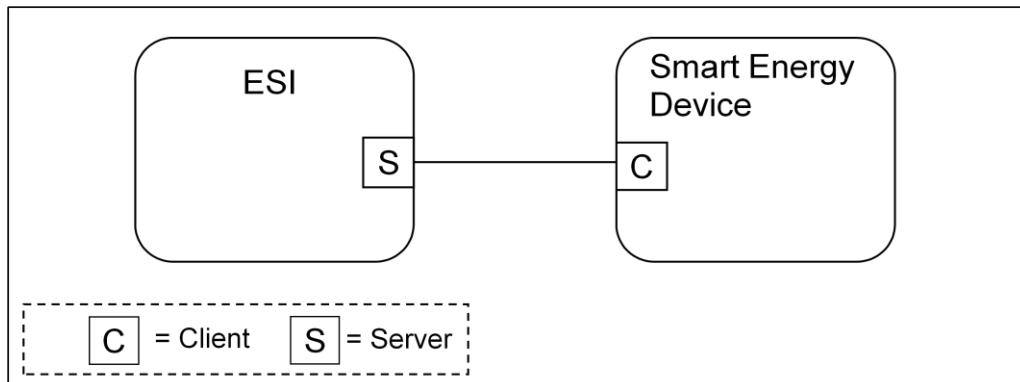
12053

12054

12055 **D.10 Device Management Cluster**12056 **D.10.1 Overview**

12057 The Device Management Cluster provides an interface to the functionality of devices within a
 12058 Smart energy Network. The cluster will support the following functions:

- 12059 • Supplier Control
- 12060 • Tenancy Control
- 12061 • Password Control
- 12062 • Event Configuration



12063
 12064 **Figure D-154– Device Management Cluster Client/Server Example**

- 12065
- 12066 **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).
- 12069 **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).
- 12072 **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.
- 12075 The Password commands provide a mechanism where a specific password located on a Smart
 12076 Energy device may be changed to a new value or reset. The server shall maintain an access

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

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

-
- 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

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

12126
12127

12128

12129 D.10.2.2.1.8 Proposed Provider Contact Details Attribute12130 An octet string containing the contact details of the Provider associated with the proposed change
12131 of supply of the commodity delivered to the premises. The attribute is capable of storing a 19
12132 character string (the first octet indicates length) encoded in UTF-8 format.12133 D.10.2.2.1.9 ReceivedProviderID Attribute12134 An unsigned 32-bit field containing a unique identifier for the commodity supplier receiving the
12135 Received energy.

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

0x0202 – 0x02FF	Reserved					
--------------------	----------	--	--	--	--	--

12164

12165 D.10.2.2.1 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 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
----------------------	------	------	-------	--------	---------	----------------------

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

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.

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
--------	---	---	-----	---	---

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

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.

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.

12320 **Tariff Type (Mandatory):** An 8-bit bitmap identifying the type of tariff published in this
12321 command. The least significant nibble represents an enumeration of the tariff type as detailed in
12322 Table D-108 (Generation Meters shall use the ‘Received’ Tariff). The most significant nibble is
12323 reserved.

12324 **Proposed Provider ID (mandatory):** An unsigned 32 bit field containing a unique identifier for
12325 the commodity provider associated with the proposed change to the supply. Depending on the
12326 *Tariff Type*, this value will be taken from either attribute D.10.2.2.1.4 or D.10.2.2.1.12.

12327 **Provider Change Implementation Time (mandatory):** A UTC Time field to indicate the
12328 date/time at which a proposed change to the provider is to be implemented. Depending on the
12329 *Tariff Type*, this value will be taken from either attribute D.10.2.2.1.6 or D.10.2.2.1.14. A
12330 *Provider Change Implementation Time* of 0xFFFFFFFF shall cause an existing but pending
12331 *Publish Change of Supplier* command with the same Current Provider ID and Issuer Event ID to
12332 be cancelled.

12333 **Proposed Provider Name (mandatory):** An octet string that denotes the name of the new
12334 commodity provider. This is dependent on the *Tariff Type* value; for Received, the parameter
12335 should match the attribute in section D.10.2.2.1.13, for all other values it should match the
12336 attribute in section D.10.2.2.1.5.

12337 **Proposed Provider Contact Details (mandatory):** An octet string that denotes the contact
12338 details of the new commodity provider. The field shall be capable of storing a 19 character string
12339 (the first octet indicates length) encoded in UTF-8 format.

12340 **Provider Change Control (mandatory):** A 32-bit mask that denotes the functions that are
12341 required to be carried out on processing of this command. See section D.10.2.2.1.7 or
12342 D.10.2.2.1.15, depending on the *Tariff Type*.

12343 D.10.2.4.2.3 When Generated

12344 The *PublishChangeofSupplier* command shall be generated from the ESI, and sent to the meter,
12345 when a change of commodity provider is required. It shall also be generated in response to a *Get*
12346 *Change of Supplier* command. The *PublishChangeofSupplier* command contains a start date/time
12347 which allows the command to be sent in advance of the changeover date.

12348 D.10.2.4.2.4 Effect on Receipt

12349 Following receipt of a *PublishChangeofSupplier* command, the meter shall only action the
12350 command at the *ProviderChangeImplementationTime*. At this point in time, the meter shall
12351 check the *Provider Change Control* field to understand what action(s) it must carry out pre and
12352 post the change.

12353

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	

12377

12378 **Password (mandatory):** An octet string of length 11 that contains the password (the first octet is
12379 the length, allowing 10 octets for the password).

12380 D.10.2.4.3.3 Effect on Receipt

12381 On receipt of this command, the client shall update the specified password.

12382

12383 D.10.2.4.4 Update SiteID Command

12384 This command is used to set the *SiteID* attribute on a meter (see D.3.2.2.4.8).

12385 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

12387 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.

12394 **SiteID Time (mandatory):** A UTC Time field to denote the time at which the update of *SiteID*
12395 will take place. A date/time of 0x00000000 shall indicate that the command should be executed
12396 immediately (comparison against a time source should NOT be made in this case). A date/time of
12397 0xFFFFFFFF shall cause an existing but pending *Update SiteID* command with the same
12398 *Provider ID* and *Issuer Event ID* to be cancelled.

12399 **Provider ID:** An unsigned 32-bit field containing a unique identifier for the commodity provider
12400 to whom this command relates.

12401 **SiteID (mandatory):** An octet string that denotes the Site ID.

12402

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

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**

12443 **Log ID:** The *Log ID* specifies the log ID of events to be updated with the new *Configuration Value* field passed in the command. The applicable values for this field are defined by bits 0-2 of
 12444 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).

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

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 Configuration 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

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	AsynchronousBillingOccurred	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

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

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 ₂ ValueReceived	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01C9	PublishCO ₂ ValueActioned	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01CA	PublishCO ₂ ValueCancelled	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x01CB	PublishCO ₂ ValueRejected	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

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 Attributes12520 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.

12522

Table D-177– Event Configuration Bitmaps

Bit	Description	
	Enumerated Value	Description
Bits 0-2	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					

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
	<i>Water: Low Pressure</i>					
	<i>Heat/Cooling: Burst Detect</i>					
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	8-bit Bitmap	0x00 – 0xFF	Read only		O
	<i>Electricity: Reserved</i>					
	<i>Gas: Reverse Flow</i>					
	<i>Water: Reverse Flow</i>					
	<i>Heat/Cooling: Flow Sensor</i>					
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

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

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	OutletTemperatureSensorFault	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

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

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

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

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

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

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 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					

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 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

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

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

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

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 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

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

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 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

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

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 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					

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

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					

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 Attributes12572 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

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.

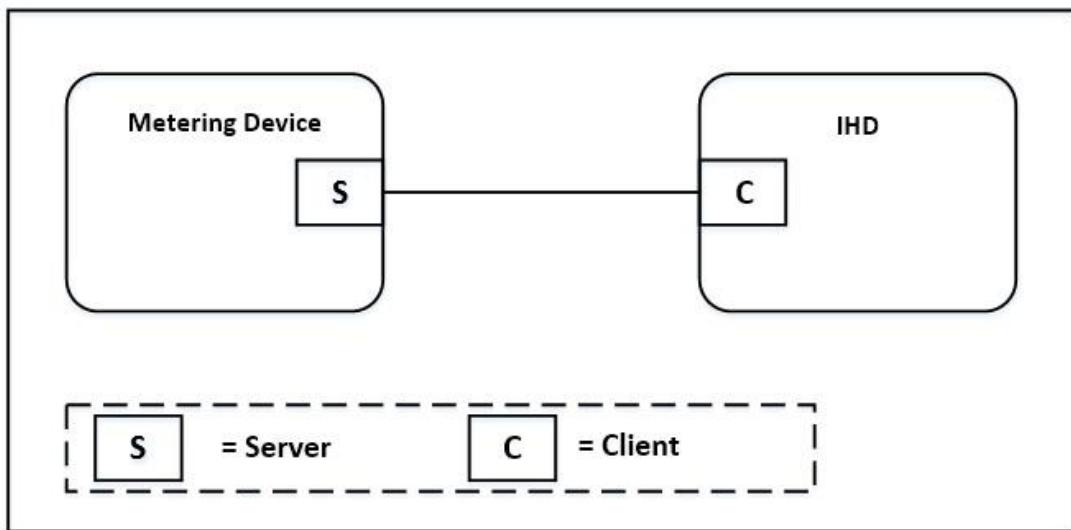
12604 If a device is unable to use a password then NO data should be shown that has been deemed to be
12605 password protected.

12606

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

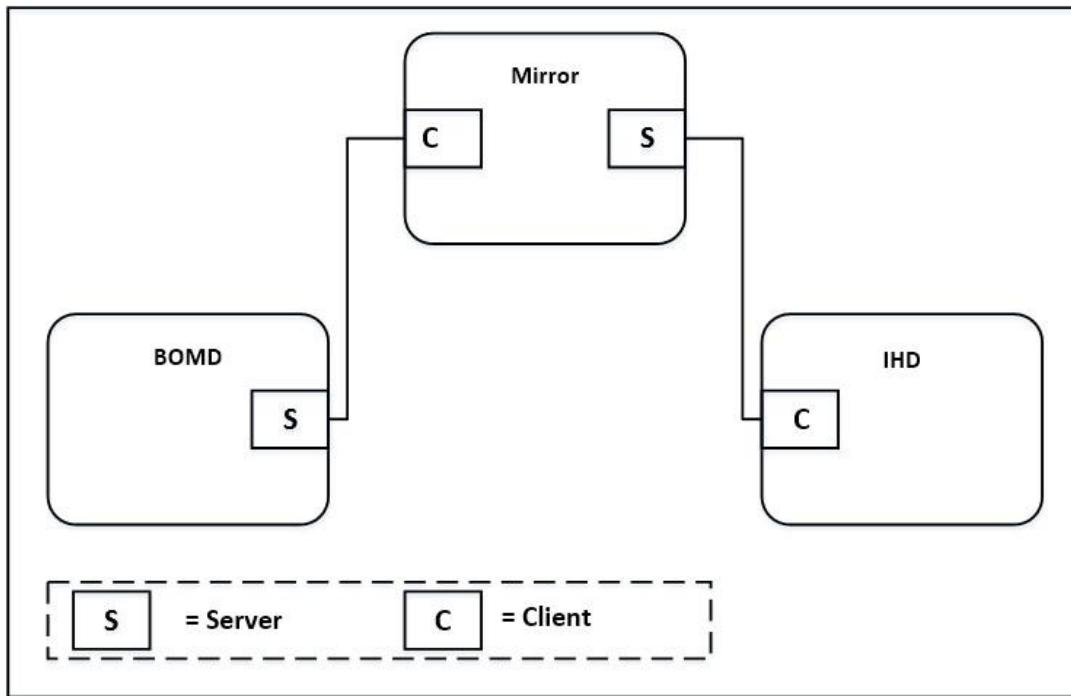


Figure D-170– Mirrored BOMD Event Cluster Client Server Example

12613

12614

12615

12616 **D.11.2 Server**

12617 **D.11.2.1 Dependencies**

- 12618 • Events carried using this cluster include a timestamp with the assumption that target devices
12619 maintain a real time clock. Devices can acquire and synchronize their internal clocks via the
12620 ZCL Time server.
- 12621 • A server device supporting this cluster should also support the Device Management Cluster
12622 in order to allow events to be configured over the air.
- 12623 • In order that Events Cluster client devices are able to receive events published from an
12624 Events Cluster server on a BOMD, the BOMD mirror should support both an Events cluster
12625 client and server. The BOMD should publish events to the mirror and the mirror should, if
12626 required (based on the control flags in the *PublishEvent* command), publish events to all
12627 bound Events Cluster client devices.
- 12628 • Events Cluster client devices wishing to receive events published from a BOMD shall bind
12629 to the Events Cluster server on the BOMD mirror.

- 12630 • A Mirror may store the events pushed from a BOMD, effectively mirroring the BOMD event
 12631 logs. The Mirror may also support the reading and clearing of event logs by Events Cluster
 12632 client devices.
- 12633 • How events are internally stored within an Events Cluster server device is out of scope of
 12634 this specification.

12635

12636 [D.11.2.2 Attributes](#)

12637 None

12638 [D.11.2.3 Commands Received](#)

12639 Table D-185 lists the cluster specific commands that are received by the server.

12640 **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

12641

12642 [**D.11.2.3.1 Get Event Log Command**](#)12643 The *GetEventLog* command allows a client to request events from a server's event logs. One or
 12644 more *PublishEventLog* commands are returned on receipt of this command.12645 The *LogID* sub-field, in conjunction with the *Event ID* field, shall provide the filtering to enable
 12646 the desired event(s) to be identified. The following examples show the usage of these 2 fields:-

- 12647 1 Get all events from the Security Event Log (Log ID = Security (4), Event ID = 0x0000)
- 12648 2 Get all events from all logs (Log ID = 0, Event ID = 0x0000)
- 12649 3 Get all occurrences of a specific event 0x1111 from all logs (Log ID = 0, Event ID =
 12650 0x1111)
- 12651 4 Get all occurrences of a specific event 0x1111 from the Security Event log (Log ID =
 12652 Security (4), Event ID = 0x1111).

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)

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 the Log ID (see Table D-186). The most significant nibble is a bitmap indicating control options (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 0x0000 is reserved to indicate all Event IDs. The Event IDs for the Smart Energy profile are 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 IDs are defined in tables Table D-176 to Table D-184.*

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

- 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 D-186). The most significant nibble is reserved.
- 12700
- 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 specified event logs. A *Clear Event Log Response* command shall be generated, indicating which event logs have been successfully cleared.
- 12703
- 12704
- 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 by the associated Event Configuration (bitmap) attribute in the Device Management cluster (see Table D-177 for further information).
- 12712
- 12713
- 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 Table D-186). The most significant nibble is reserved.
- 12718

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 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.

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

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.

12761 **Event ID:** The *Event ID* specifies a particular event. If event configuration is supported (via the
12762 Device Management cluster), ZigBee-specified Event IDs are as defined in Table D-176 to Table
12763 D-184.

12764 Event Time: The timestamp of the event occurrence in UTC format.

12765 **Event Data:** A variable length octet string array used to hold additional information captured
12766 when the event occurred. The first element (element 0) of the array indicates the length of the
12767 string, NOT including the first element. This field should contain a single octet of 0x00 when
12768 ‘minimal information’ is requested in the associated *Get Event Log* command (see D.11.2.3.1.2
12769 for further details).

12770

12771 D.11.2.4.3 Clear Event Log Response Command

12772 This command is generated on receipt of a *Clear Event Log Request* command.

12773 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

12775 D.11.2.4.3.2 Payload Details

12776 **ClearedEventsLogs (mandatory):** This 8-bit BitMask indicates which logs have been cleared,
12777 as detailed in Table D-191.

12778 Note: It is understood that certain markets may require that event logs cannot be cleared; this
12779 BitMask provides a method for the server device to indicate which logs have been successfully
12780 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

	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.

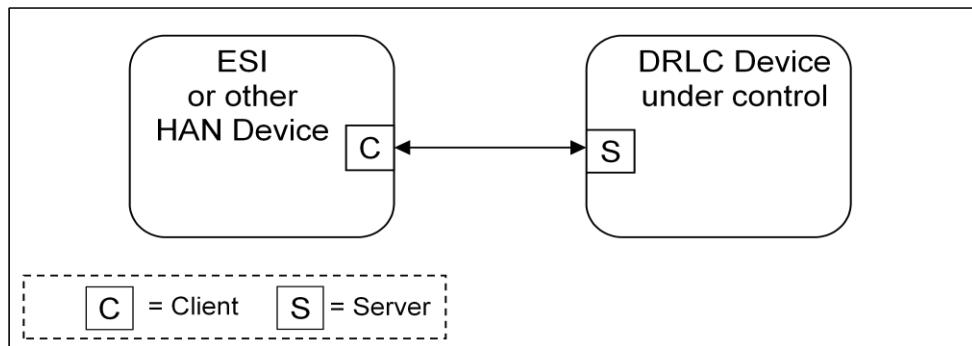
12795

12796 **D.12 Energy Management Cluster**

12797 *Note: The Energy Management Cluster description in this revision of this specification is
12798 provisional and not certifiable. This feature set may change before reaching certifiable
12799 status in a future revision of this specification.*

12800 **D.12.1 Overview**

12801 This cluster provides a way of modifying DRLC events, energy consumption behaviour and
12802 querying the status of DRLC events.

12803
12804 **Figure D-177– Energy Management Cluster**
12805

12806 Note that the ESI is defined as the Client. The DRLC device is a Server in this case; it holds the
12807 attributes and receives commands.
12808

12809 **D.12.2 Server**12810 **D.12.2.1 Dependencies**

12811 A server device shall support the DRLC cluster as a client.

12812 **D.12.2.2 Attributes**12813 **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 0xFFFFFFFF	Read Only	0xFFFFFFFF	M

0x0002	<i>CurrentEventStatus</i>	8-bit BitMap	0x00 to 0xFF	Read Only	0x00	M
0x0003	<i>ConformanceLevel</i>	Unsigned 8-bit Integer	0x00 to 0x07	Read/ Write or Read Only	0x00	M
0x0004	<i>MinimumOffTime</i>	Unsigned 16-bit Integer	0x0000 to 0x0258 (10 minutes) or 0xFFFF	Read/ Write or Read Only	TBD	M
0x0005	<i>MinimumOnTime</i>	Unsigned 16-bit Integer	0x0000 to 0x0258 (10 minutes) or 0xFFFF	Read/ Write or Read Only	TBD	M
0x0006	<i>MinimumCyclePeriod</i>	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.

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 **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.12.2.3 **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

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])

12882 and the *OnOff* attribute was set to OFF. It would also be False if the device was a PCT, the
 12883 current temperature was lower than the set point and the device was in cooling mode. Note that,
 12884 if an event is not active, this value may not be reliable. Support is optional; if unsupported, the bit
 12885 shall be set.

12886 **D.12.2.2.4 Conformance Level Attribute**

12887 This is the minimum criticality level of a DRLC event that the device will observe. Events with a
 12888 criticality level lower than the *Conformance Level* will be opted out. If a change in the
 12889 *Conformance Level* results in a currently running event being opted out (or in), a
 12890 *ReportEventStatus* message must be generated. If the event(s) has(have) not started, the device
 12891 may send a *ReportEventStatus* immediately or when the event starts. The attribute shall be
 12892 read/write if supported, with a valid range of 1 -7, or 0 and read-only if unsupported.

12893 **D.12.2.2.5 MinimumOffTime, MinimumONTime & MinimumCyclePeriod Attributes**

12894 These attributes are measured in seconds and used to determine the duty cycling times. The
 12895 *MinimumOffTime* and *MinimumOnTime* may also be enforced when starting or ending a DRLC
 12896 event, or when the On/Off cluster opens or closes the relay. From these values, the DutyOnTime
 12897 and DutyOffTime times are calculated.

12898 *MinimumCyclePeriod* is the shortest cycling period allowed for duty cycling, while
 12899 *MinimumOnTime* and *MinimumOffTime* are the shortest times the device should be allowed on or
 12900 off. The on/off minimums are to protect devices that can be damaged by being on or off for short
 12901 time periods, while the *MinimumCyclePeriod* prevents the relay from cycling too quickly. The
 12902 attribute values shall be 0xFFFF and read-only if unsupported, or read/write if supported. A
 12903 *MinimumCyclePeriod* of 0 indicates that duty cycling shall be disabled; an event calling for any
 12904 duty cycling will then just turn the device off for the duration of the event.

12905

12906 **Algorithm for calculating DutyOnTime and DutyOffTime**

12907
 12908 DutyOnTime: the amount of time in a duty cycle period that the device is on
 12909 DutyOffTime:the amount of time in a duty cycle period that the device is off
 12910
 12911 DutyOnTime = MinimumCyclePeriod * dutyCycleApplied / 100
 12912 DutyOffTime = MinimumCyclePeriod - DutyOnTime
 12913 //NOTE except for dutyCycleApplied near 99% or 1% we are likely done
 12914
 12915 If (DutyOnTime < MinimumONTime)
 12916 DutyOnTime = MinimumONTime
 12917 DutyOffTime = MinimumONTime * (100 - dutyCycleApplied) / dutyCycleApplied
 12918
 12919 If (DutyOffTime < MinimumOffTime)
 12920 DutyOffTime = MinimumOffTime
 12921 DutyONTime = MinimumOffTime * dutyCycleApplied/ (100 - dutyCycleApplied)

- 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.

Table D-195– Received Command IDs for the Energy Management Server

Command Identifier Field Value	Description	Mandatory/Optional
0x00	<i>ManageEvent</i>	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

Figure D-178– Manage Event Command Payload

- 12934
- 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 Enrollment Group*
- 12944 field can be used in conjunction with the *Device Class* bits. It provides a mechanism to direct the

12945 *Manage Event* command to groups of devices. If the requested UEG is non-zero and does not
12946 match the UEG of the receiving device, the command should be ignored.

12947 **Action(s) Required:** Bit encoded field indicating the action(s) to be carried out in regard to the
12948 associated event. Bits, set individually or in combination, shall be provided for the following
12949 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

12951

12952 If the *Manage Event* command is valid, a *Report Event Status* command will be returned
12953 regardless of whether any bits are set.

12954 Notes:

- 12955 1 If the load control event is mandatory and the device is Smart Energy compliant, requests to
12956 opt-out shall be ignored. Opt-in shall always be honored.

12957 2 If the device does not support duty cycling, or the load control event did not request duty
12958 cycling, requests to modify duty cycling should be ignored, however a *Report Event Status*
12959 command should still be sent.

12960 3 If the *ManageEvent* command action was to opt-out or opt-in, the *Event Status* field in the
12961 associated *Report Event Status* response shall be 0x04 (User Opt-out) or 0x05 (User Opt-in).
12962 For all other actions, the *Event Status* field in the associated *Report Event Status* response
12963 shall be set to 0x01 (if the load control event has not yet started), 0x02 (if the event is
12964 running and opted in) or 0x04 (if the event is running but currently opted out; this is to allow
12965 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.

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	<i>ReportEventStatus</i>	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:

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

13014 **D.13 MDU Pairing Cluster**

13015 *Note: The MDU Pairing Cluster description in this revision of this specification is provisionalary and not certifiable. This feature set may change before reaching certifiable status in a future revision of this specification.*

13018 **D.13.1 Overview**

13019 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.

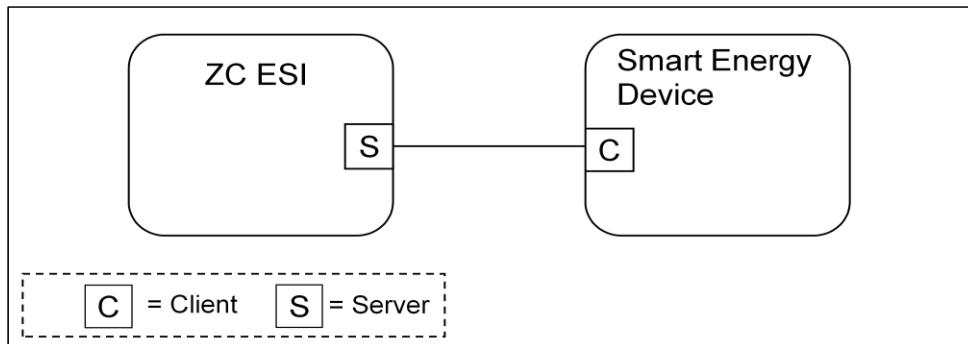


Figure D-180– MDU Pairing Cluster

13023
13024
13025

13026 **D.13.2 Server**

13027 **D.13.2.1 Dependencies**

13028 Support for ZCL Data Types.

13029 **D.13.2.2 Attributes**

13030 There are no attributes for the MDU Pairing cluster server.

13031 **D.13.2.3 Commands Generated**

13032 The command IDs generated by the MDU Pairing cluster server are listed in Table D-198.

13033 **Table D-198– MDU Pairing Server Commands**

Command Identifier Field Value	Description	Mandatory / Optional
0x00	Pairing Response	O

13034

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).

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 it 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.

Table D-199– MDU Pairing Client Commands		
Command Identifier Field Value	Description	Mandatory / Optional
0x00	Pairing Request	M

Table D-199– MDU Pairing Client Commands		
Octets	4	8
Data Type	Unsigned 32-bit Integer	IEEE Address

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

Field Name	Local Pairing Information Version (M)	EUI64 of Requesting Device (M)
------------	---------------------------------------	--------------------------------

Figure D-182– Pairing Request Command Payload

13086

13087 D.13.3.3.1.2 Payload Details

13088 **Local Pairing Information Version (mandatory):** Identifies the version of pairing information
 13089 currently held on the requesting device. The TC ESI shall use this information to determine if
 13090 new pairing information is to be returned. A device that does not yet have local pairing
 13091 information shall set this field to zero (0x00000000).

13092 **EUI64 of Requesting Device (mandatory):** Identifies the MAC address of the requesting
 13093 device.

13094 D.13.3.3.1.3 When Generated

13095 A device should send this command to the TC ESI of a network that it is in the process of
 13096 joining. It should also periodically send the command to the TC ESI in order to determine if
 13097 further devices have joined its ‘virtual HAN’ or that the devices constituting the ‘virtual HAN’
 13098 have changed (*Pairing Information Version* has been updated).

13099 D.13.3.3.1.4 Effect on Receipt

13100 If the command is received by a TC ESI that is not operating in a MDU, or the receiving ESI is
 13101 not the TC ESI, a ZCL Default Response shall be returned with a status of
 13102 UNSUP_CLUSTER_COMMAND (the joining device should then follow normal (traditional)
 13103 service discovery and binding procedures).

13104 If the command contains a *Local Pairing Information Version* that matches the version of pairing
 13105 information for the requesting device that is held by the TC ESI, a ZCL Default Response shall
 13106 be returned with a status of WAIT_FOR_DATA (the same response should be returned if data is
 13107 not yet available on the TC ESI for the requesting device).

13108 If the command is received by a TC ESI that is controlling a MDU, the TC ESI should return a
 13109 *Pairing Response* command indicating the devices that will constitute the ‘virtual HAN’ for the
 13110 household in which the requesting device is to operate. A joining device should then follow
 13111 normal (traditional) service discovery and binding procedures, but only with those devices
 13112 identified in the response command.

13113 Note that this command should not be recognized by an older Smart Energy TC ESI; this should
 13114 result in that TC ESI returning a ZCL Default Response with a status of
 13115 UNSUP_CLUSTER_COMMAND.

13116

13117 Annex E RULES AND GUIDELINES FOR OVERLAPPING EVENTS

13118
13119 This section describes multiple scenarios that Demand Response and Load Control
13120 devices may encounter over the Smart Energy network. The examples describe situations of
13121 overlapping events that are acceptable and where overlapping events that will be
13122 superseded due to conflicts.

13123 **E.1 Definitions**

13124 **Start Time** – “Start Time” field contained within the Load Control Event packet indicating
13125 when the event should start. Please note, a “Start Time” value of 0x00000000 denotes
13126 “now” and the device should use its current time as the “Start Time”.

13127 **Duration** – “Duration” field contained within the Load Control Event packet indicating how
13128 long the event should occur.

13129 **End Time** – Time when Event completes as calculated by adding *Duration* to *Start Time*.

13130 **Scheduled Period** - Represents the time between the *Start Time* and the *End Time* of the event.

13131 **Effective Start Time** - Represents time at which a specific device starts a load control event
13132 based on the *Start Time* plus or minus any randomization offsets.

13133 **Effective End Time** - Represents time at which a specific device ends a load control event
13134 based on the *Start Time* plus *Duration*, plus or minus any randomization offsets.

13135 **Effective Scheduled Period** - Represents the time between the *Effective Start Time* and the
13136 *Effective End Time*.

13137 **Overlapping Event** - Defined as an event where the *Scheduled Period* covers part or all of an
13138 existing, previously scheduled event.

13139 **Successive Events** - Defined as two events where the scheduled *End Time* of the first event is
13140 equal the *Start Time* of a subsequent scheduled event.

13141 **Nested Events** - Defined as two events where the scheduled *Start Time* and *End Time* of the
13142 second event falls during the *Scheduled Period* of the first scheduled event and the second event
13143 is of shorter duration than the first event.

13144 **E.2 Rules and Guideline**

13145 The depicted behaviors and required application management decisions are driven from the
13146 following guidance and rule set:

- 13147 **1** Upstream Demand Response/Load Control systems and/or the ESI shall prevent
13148 mismanaged scheduling of *Overlapping Events* or *Nested Events*. It is recognized Upstream
13149 Demand Response/Load Control systems and/or the ESI will need to react to changing
13150 conditions on the grid by sending *Overlapping Events* or *Nested Events* to supersede previous
13151 directives. But those systems must have the proper auditing and management rules to prevent
13152 a cascading set of error conditions propagated by improperly scheduled events.
- 13153 **2** When needed, Upstream Demand Response/Load Control systems and/or the ESI may
13154 resolve any event scheduling conflicts by performing one of the following processes:
 - 13155 **a** Canceling individual events starting with the earliest scheduled event and re- issuing a new
13156 set of events.
 - 13157 **b** Canceling all scheduled events and re-issuing a new set of events.
 - 13158 **c** Sending *Overlapping Events* or *Nested Events* to supersede previous directives.
13159 It is recommended that process 2.c is used for most situations since it can allow a smoother
13160 change between two sets of directives, but no way does it negate the responsibilities identified
13161 in rule #1.
- 13162 **3** When an End Device receives an event with the *End Time* in the past (*End Time* < Current
13163 Time), this event is ignored and a *Report Event Status* command is returned with the Event
13164 Status set to 0xFB (Rejected - Event was received after it had expired).
- 13165 **4** When an End Device receives an event with a *Start Time* in the past and an *End Time* in the
13166 future ((*Start Time* < Current Time) AND (*End Time* > Current Time)), the event is processed
13167 immediately. The Effective *Start Time* is calculated using the Current Time as the *Start Time*.
13168 Original *End Time* is preserved.
- 13169 **5** Regardless of the state of an event (scheduled or executing), when an *End Device* detects an
13170 *Overlapping Event* condition the latest *Overlapping Event* will take precedence over the
13171 previous event. Depending on the state of the event (scheduled or executing), one of the
13172 following steps shall take place:
 - 13173 **a** If the previous event is scheduled and not executing, the End Device returns a *Report*
13174 *Event Status* command (referencing the previous event) with the Event Status set to 0x07
13175 (The event has been superseded). After the *Report Event Status* command is successfully
13176 sent, the End Device can remove the previous event schedule.

13177 b If the previous event is executing, the End Device shall change directly from its current
13178 state to the requested state at the *Effective Start Time* of the *Overlapping Event* (Note:
13179 Rule #4 effects *Effective Start Time*). The End Device returns a *Report Event Status*
13180 command (referencing the previous event) with the Event Status set to 0x07 (the event has
13181 been superseded).

13182 **6** Randomization ***shall not*** cause event conflicts or unmanaged gaps. To clarify:

13183 a When event starting randomization is requested, time periods between the *Start Time* of an
13184 event and the *Effective Start Time* a device should either maintain its current state or apply
13185 changes which contribute to energy saving. Preference would be to maintain current state.

13186 b When event ending randomization is used and the *Effective End Time* overlaps the
13187 *Effective Start Time* of a *Successive Event*, the *Effective Start Time* takes precedence.
13188 Events are not reported as superseded, End devices should report event status as it would a
13189 normal set of *Successive Events*.

13190 c It is recommended devices apply the same Start and Stop Randomization values for
13191 consecutive events to help prevent unexpected gaps between events.

13192 d Devices ***shall not*** artificially create a gap between *Successive Events*.

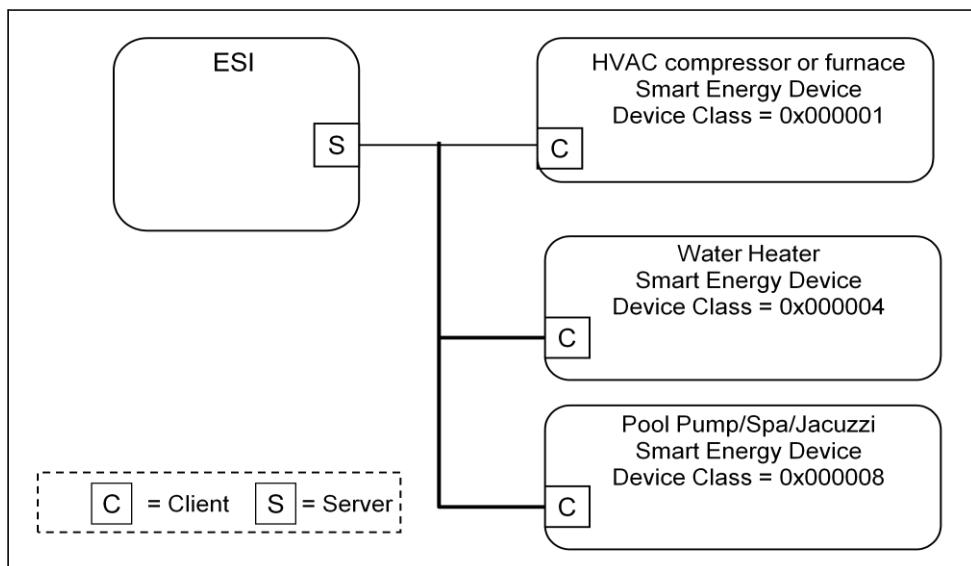
13193 **7** It is permissible to have gaps when events are not *Successive Events* or *Overlapping Events*.

13194 **8** If multiple device classes are identified for an event, future events for individual device
13195 classes (or a subset of the original event) that cause an *Overlapping Event* will supersede the
13196 original event strictly for that device class (or a subset of the original event). Note: Rule #5
13197 applies to all *Overlapping Events*.

13198
13199

13200 **E.3 Event Examples**

13201 Smart Energy devices which act upon Demand Response and Load Control events shall use the
13202 following examples for understanding and managing overlapping and superseded events.
13203 Within those examples, references to multiple device classes will be used. Figure E-1 depicts
13204 a representation of those devices in a Smart Energy network.



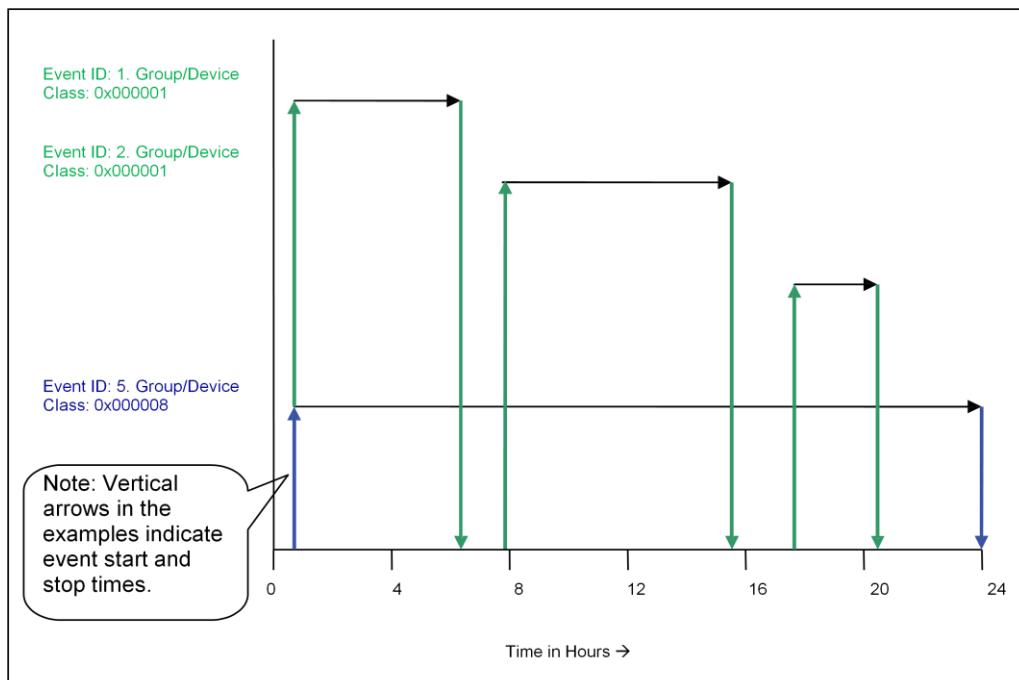
13205

13206

Figure E-1– Smart Energy Device Class Reference Example

13207

- 13208 **E.3.1 Correct Overlapping Events for Different Device Classes**
- 13209 Figure E-2 depicts a correct series of DR/LC event for device class of 0x000001 (reference for
13210 the BitMap definition) with an event scheduled for another device class during the same period.

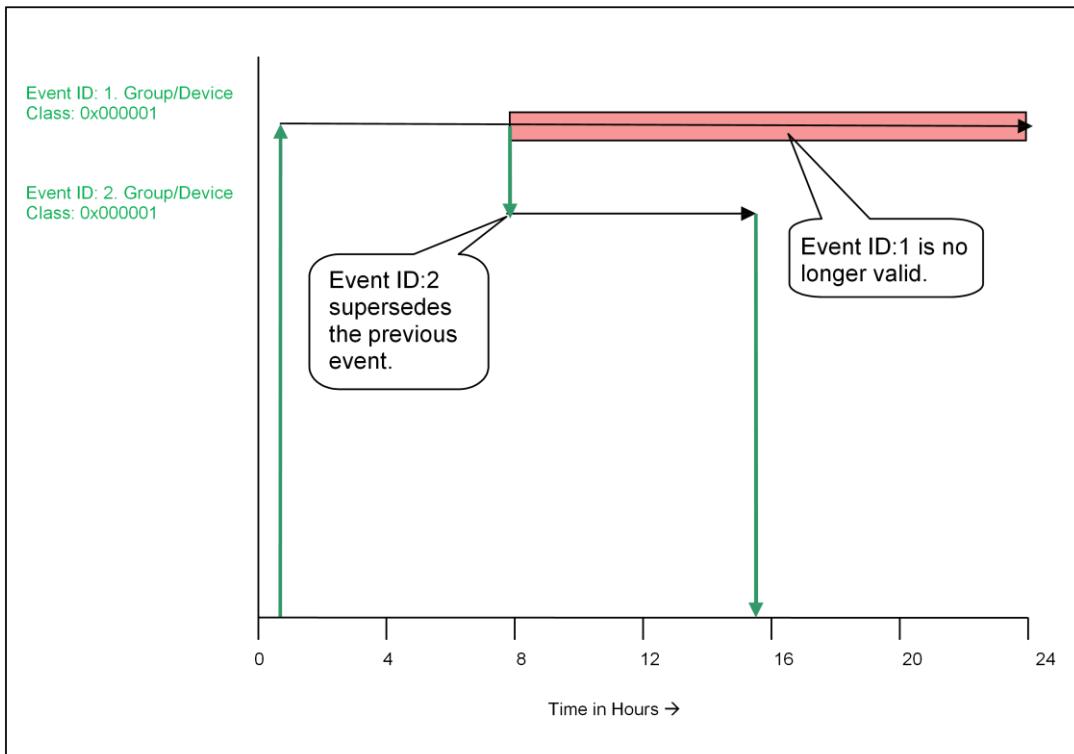


- 13211 **Figure E-2– Correctly Overlapping Events**
- 13212 In Figure E-2, Device Class 0x000001 receives a sequence of 3 unique DR/LC events to be
13213 scheduled and acted upon. During this same 24 hour period, Device Class 0x000008 receives
13214 one scheduled DR/LC event that spans across the same time period as the events scheduled for
13215 Device Class 0x00000001. Because both Device Classes are unique, there are no conflicts due to
13216 Overlapping Events.
- 13217

13218

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

13228

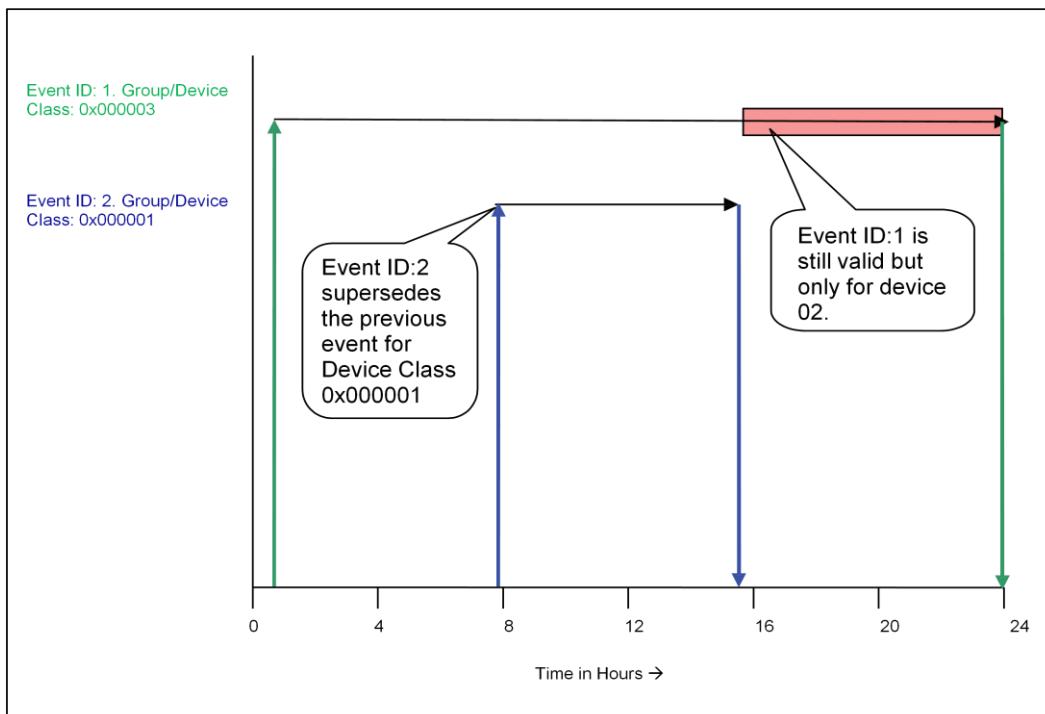
E.3.3 Superseding Events for Subsets of Device Classes

13229

13230

13231

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.



13232

13233

Figure E-4—Superseded Event for a Subset of Device Classes

13234

13235

13236

13237

13238

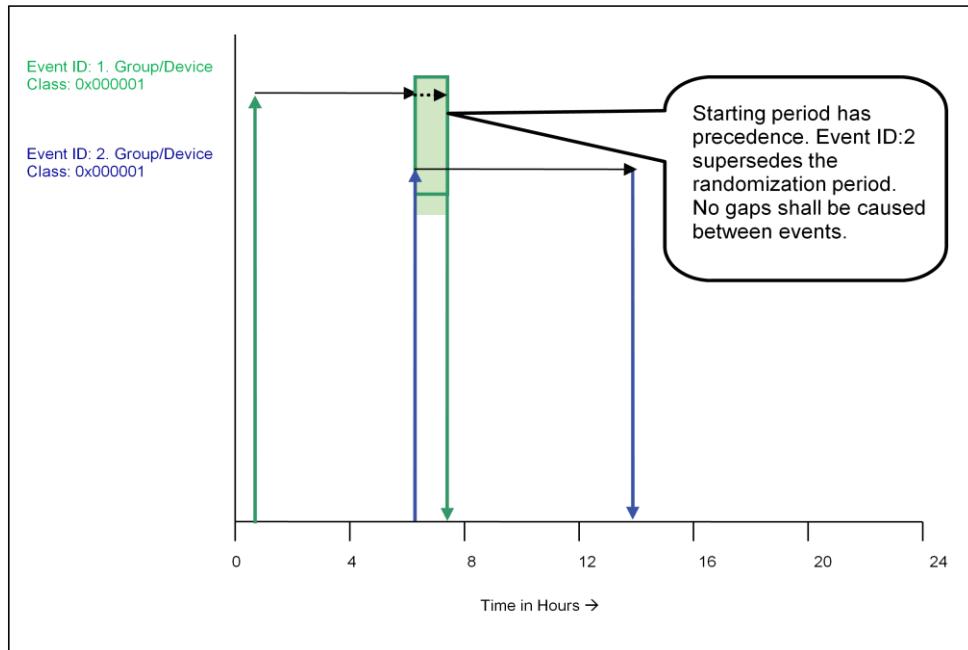
In Figure E-4, Device Class 0x000003 receives DR/LC Event ID#1 setup for a 24 hour *Scheduled Period*, which is targeted for both Device Class 0x000002 and 0x000001 (OR'ed == 0x000003). In the example, Event ID#2 is issued only for Device Class 0x000001, invalidating the remainder of Event ID#1 for that device class. DR/LC Event ID#1 is still valid for Device Class 0x000002, which in the example should run to completion.

13239

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

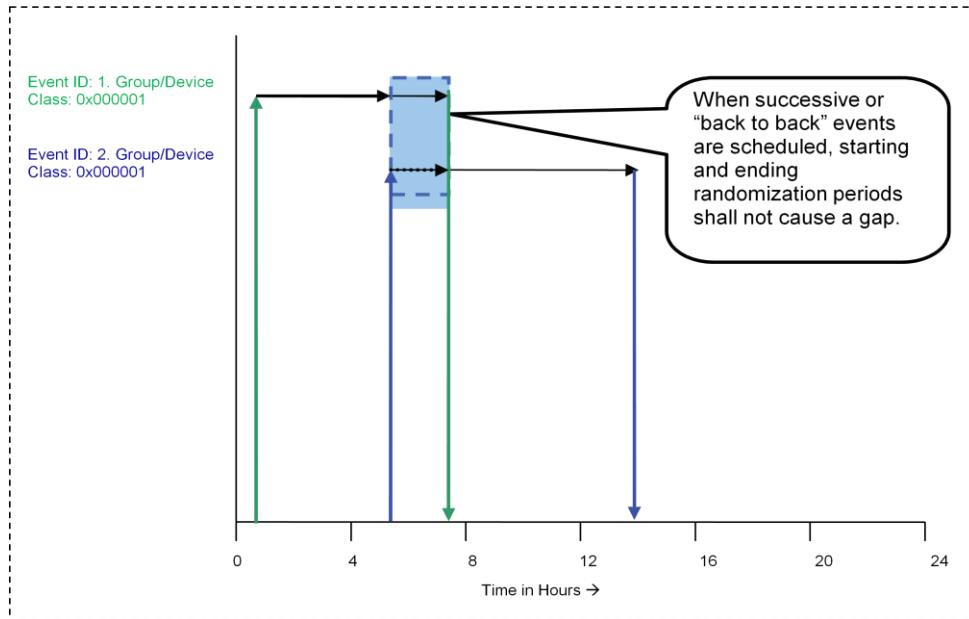
13244
13245**Figure E-5– Ending Randomization Between Events**

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

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

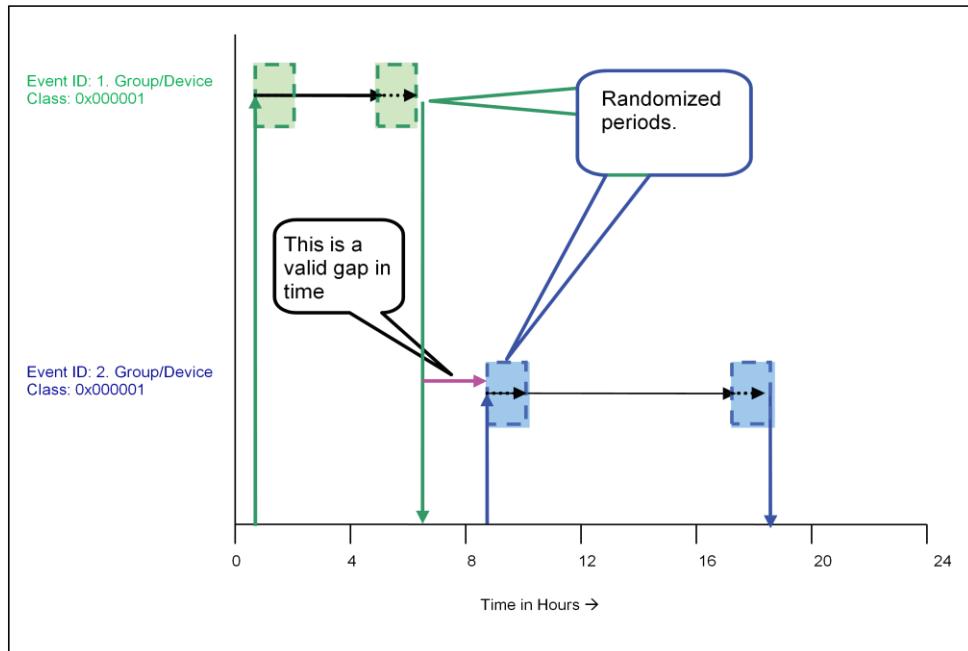
Figure E-6– Start Randomization Between Events

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 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

13265 **E.3.6 Acceptable Gaps Caused by Start and Stop Randomization of Events**

13266 Figure E-7 below depicts an acceptable gap between two scheduled DR/LC events for device
 13267 class of 0x000001 (reference for the BitMap definition) using both starting and ending
 13268 randomization with both events.



13269 **Figure E-7– Acceptable Gaps with Start and Stop Randomization**

13270 Figure E-7 above, Device Class 0x000001 receives a DR/LC Event ID#1 with both a starting
 13271 and ending randomization setting (please refer to sub-clause D.2.2.3.1.1 for more detail). A
 13272 second DR/LC Event ID#2 is also issued with both a starting and ending randomized setting.
 13273 The primary configuration to note in this example is the *Effective End Time* of DR/LC Event
 13274 ID#1 completes well in advance of the *Effective Start Time* of DR/LC Event ID#2. In this
 13275 scenario, regardless of randomization a gap is naturally created by the scheduling of the events
 13276 and is acceptable.

13277

Annex F JOINING PROCEDURE USING PRE-CONFIGURED TRUST CENTER LINK KEYS

13279
13280
13281

13282 The secure join procedure is detailed as follows:

- 13283 • The secured joining procedure is as stated in [B3] Section 4.6.3.2.3. The case used in the
13284 Smart Energy application is the “Pre-configured trust center link key and address”
- 13285 • In [B3] Section 4.6.3.2.3.2, in the case of “Pre-configured Trust Center Link Key”, the
13286 joining device waits for the APSME-TRANSPORT-KEY.Indication. The frame is
13287 encrypted/authenticated with the key-transport key according to the methodologies specified
13288 in sections 4.4.1.1 and 4.5.3 of the ZigBee specification r17, which describe the key-
13289 transport keys and their association with link keys, in this case the pre-configured trust center
13290 link key. The source address will be that of the Trust Center. The key transported will be the
13291 NWK Key Key type == 0x01.
- 13292 • When the trust center sends the tunneled *Transport Key* command, the Extended Nonce bit
13293 on the Auxiliary Frame Header must be set to 1 on the Transport Key frame from the Trust
13294 Center to the joining child as described in [B3] Section 4.5.1. The Trust Center must also
13295 insert its long address into the Source Address field of the Auxiliary Frame Header since that
13296 information will be needed at the child to decrypt the *Transport Key* command.
- 13297 • Sub-clause 5.4 of this document calls out two cases for secured join: pre- configured link
13298 keys and temporary link keys. The joining device and trust center perform the same join
13299 operation in both cases. The only difference is how the joining device and trust center treat
13300 the initial key material (either using it directly as the pre-configured link key or hashing with
13301 some data like the long address of the joining device at application level first, see Annex E
13302 for this method). From the perspective of the security joining process what happens
13303 afterwards is the secure join procedure is the same.
- 13304 • In either case called out in sub-clause 5.4 of this document, the joining device is
13305 authenticated using the [B3] Section 4.6.3.2.3.2 procedure or leaves if the security timeout
13306 expires. If authenticated, the key delivered via the APSME- TRANSPORT-KEY.indication
13307 in [B3] Section 4.6.3.2.3.2 is the same for either case called out in the AMI specification sub-
13308 clause 5.4 (no matter how the application determined the pre-configured link key).

13309 In terms of the message exchange between the child and trust center in performing the secure join
13310 procedure, the following is employed:

- 13311 1 Child joining device uses NLME-JOIN.request to parent. Parent sends an APSME-
13312 UPDATE-DEVICE.request to the Trust Center on behalf of the child to the Trust Center.

13313 APSME-UPDATE-DEVICE.request is transported encrypted/authenticated with the NWK
13314 key that the parent has

13315 **2** Upon receipt at the trust center, the trust center must perform the following processing:

13316 a Validity check of the child's address to determine if a trust center link key exists between
13317 the trust center and the address provided by the joining child.

13318 b If the child has the trust center as its parent, the APSME-TRANSPORT- KEY.request is
13319 sent directly to the child encrypted with the key-transport key derived from the trust center
13320 link key known to the child device and the trust center, ELSE

13321 i. If the child does not have the trust center as its parent, the APSME- TRANSPORT-
13322 KEY command frame is encrypted using the key-transport key derived from the
13323 trust center link key shared between the child and the trust center.

13324 c The resulting encrypted payload is sent to the child using the *APS Tunnel* command. The
13325 *APS Tunnel* command and its (already encrypted) payload is encrypted using the NWK
13326 key from the trust center to the child's parent. On the final hop, the child's parent will
13327 perform the following processing according to [B3] Section 4.6.3.7.2:

13328 i. The parent sends the contents within the *APS Tunnel* command to the child without
13329 network layer encryption. The message from the parent to the joining child is an APS
13330 encrypted transport key command using the key-transport key derived from the trust
13331 center link key.

13332 Here are the details on the message that is routed from the trust center to the joining device's
13333 parent via the *Tunnel* command:

- 13334 • NWK Data Frame (Dest: Parent)
- 13335 • APS Header (Command)
- 13336 • APS Command Frame (Tunnel)
- 13337 • Dest EUI: Child
- 13338 • Tunnel Payload
- 13339 • APS Header
- 13340 • APS Auxiliary Header
- 13341 • Encrypted Payload
- 13342 • APS Command Frame (Transport Key)

13343 Here are the details on the message that is routed from the joining device's parent to the joining
13344 child:

- 13345 • NWK Data Frame (added by parent, Dest: child)
- 13346 • APS Header (from Tunnel Payload)
- 13347 • APS Auxiliary Header (from Tunnel Payload)
- 13348 • Encrypted Payload (from Tunnel Payload)
- 13349 • APS Command Frame (Transport Key)

13350 The message to the child from the parent is identical if the device joins directly to the Trust
13351 Center.

13352 As a note on the final hop contents of the payload:

- 13353 • The last hop of the APME-TRANSPORT-KEY message from parent to joining child has NO
13354 network layer encryption, but does have application layer encryption
- 13355 • Thus: There will be no NWK auxiliary header, but there will be an APS auxiliary header
- 13356 • The APS auxiliary header will have the Key Identifier Sub-Field set to 0x02 == A key-
13357 transport key (see [B3] Section 4.5.1.1.2)
- 13358 • The APS frame will be encrypted with the key-transport key derived from the pre-configured
13359 trust center link key. The pre-configured trust center link key must be part of the
13360 apsDeviceKeyPairSet in the AIB of the joining device and also known to the trust center.
- 13361 • The resulting APS frame from the parent to the joining child is the APS- TRANSPORT-KEY
13362 message encrypted with the key-transport key derived from the trust center link key delivered
13363 with the key type of key-transport key (0x02).
- 13364 • Per [B3] Section 4.4.3.2, the KeyType field will be set to (0x01) == Network Key
- 13365 • The TransportKeyData will be the active network key and sequence number
- 13366 • The joining device must set the network key and sequence number in its NWK Information
13367 Block.
- 13368 • The device is then joined and authenticated.

13369
13370

13371
13372
13373
13374
13375
13376
13377
13378
13379
13380
13381
13382

--- END OF DOCUMENT ---