

1



**ZigBee®
Alliance**

2

3

4

5 **DRAFT ZIGBEE SMART
6 ENERGY STANDARD
7 (Editors' Copy)**

8

9

10

11 **ZigBee Smart Energy
12 Standard**

13 ZigBee Profile: 0x0109

14 Revision t.b.d.

15 Version 1.2a (0.9)

16

17

18

19

20 **24th June, 2014**

21

22

23 **Notice of Use and Disclosure**

24
25 Copyright © ZigBee Alliance, Inc. (2007-2014). All rights Reserved. The information
26 within this document is the property of the ZigBee Alliance and its use and disclosure
27 are restricted.

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

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

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

50 ZigBee Alliance, Inc.
51 2400 Camino Ramon, Suite 375
52 San Ramon, CA 94583, USA

Table of Contents

53	Notice of Use and Disclosure.....	2
54	Table of Contents	3
55	List of Tables.....	17
56	List of Figures	23
57	Participants	29
58	Document History	31
59	1 Introduction	32
60	1.1 Scope	32
61	1.2 Purpose	32
62	1.3 Provisional Features	32
63	2 References	33
64	2.1 References	33
65	2.1.1 ZigBee Alliance Documents	33
66	2.1.2 External Reference Documents	33
67	3 Definitions.....	35
68	3.1 Conformance Levels.....	35
69	3.2 ZigBee Definitions	35
70	3.3 Smart Energy Definitions	36
71	4 Acronyms And Abbreviations.....	38
72	5 Profile Description	40
73	5.1 A ZigBee Smart Energy Network.....	40
74	5.2 ZigBee Stack Profile	43
75	5.2.1 MAC Data Polling (NLME_Requests).....	44
76	5.2.2 Application Level Queries.....	44
77	5.2.3 ZigBee Coordinator and Trust Center Recommendations	45
78	5.3 Startup Attribute Set (SAS).....	45
79	5.3.1 Startup Parameters	45
80	5.3.2 Join Parameters	47
81	5.3.3 Security Parameters	48
82	5.3.4 End Device Parameters	48
83	5.3.5 Link Status Parameters	49
84	5.3.6 Concentrator Parameters	49
85	5.3.7 APS Transport Parameters.....	49
86	5.3.8 APS Fragmentation Parameters	50
87	5.3.9 Binding Parameters	50
88	5.4 Smart Energy Profile Security.....	51
89	5.4.1 Joining with Preinstalled Trust Center Link Keys.....	51
90	5.4.2 Re-Joining a Secured Network.....	53
91	5.4.3 Devices Leaving the Network.....	60
92	5.4.4 Updating the Network Key	60
93	5.4.5 Updating the Link Key	60
94	5.4.6 Cluster Usage of Security Keys	63

96	5.4.7	Key Establishment Related Security Policies	65
97	5.4.8	Security Best Practices	68
98	5.5	Commissioning	71
99	5.5.1	Forming the Network (Start-up Sequence).....	71
100	5.5.2	Support for Commissioning Modes	72
101	5.5.3	Commissioning Documentation Best Practices	72
102	5.5.4	Commissioning Procedure for Different Network Types.....	72
103	5.5.5	ZigBee Smart Energy Joining, Service Discovery, and Device Binding Requirements.....	74
104	5.6	Federated Trust Center Application Guidelines	80
105	5.7	Multiple ESI Application Guidelines	80
106	5.7.1	Overview	80
107	5.7.2	Device Behavior.....	81
108	5.8	Other Smart Energy Profile Requirements and Best Practices	83
109	5.8.1	Preferred Channel Usage	83
110	5.8.2	Broadcast Policy	84
111	5.8.3	Frequency Agility	84
112	5.8.4	Key Updates	84
113	5.9	Coexistence and Interoperability with HA Devices.....	84
114	5.10	Device Descriptions.....	84
115	5.11	ZigBee Cluster Library (ZCL)	85
116	5.12	Cluster List and IDs	86
117	5.12.1	ZCL General Clusters	88
118	5.13	Coexistence with devices using other Profiles	89
119	5.13.1	Requirements for the Smart Energy Trust Center	89
120	5.13.2	Requirements for Smart Energy Devices	89
121	5.13.3	Requirements for Devices from other Profiles.....	90
122	6	Device Specifications	92
123	6.1	Common Clusters	92
124	6.1.1	Optional Support for Clusters with Reporting Capability.....	92
125	6.1.2	Manufacturer-Specific Clusters.....	92
126	6.1.3	Cluster Usage Restrictions	92
127	6.1.4	Identify Cluster Best Practices	93
128	6.1.5	Inter-PAN Communication	93
129	6.2	Feature and Function Description	93
130	6.3	Smart Energy Devices.....	95
131	6.3.1	Energy Service Interface.....	95
132	6.3.2	Metering Device.....	96
133	6.3.3	In-Home Display Device.....	98
134	6.3.4	Programmable Communicating Thermostat (PCT) Device	99
135	6.3.5	Load Control Device	99
136	6.3.6	Range Extender Device	100
137	6.3.7	Smart Appliance Device.....	100

139	6.3.8	Prepayment Terminal Device	101
140	6.3.9	Physical Device.....	102
141	6.3.10	Remote Communications Device	103
142	Annex A	Candidate ZCL Material For Use with this Profile.....	105
143	A.1	New Status Enumeration	105
144	A.2	New Attribute Reporting Status Indication	105
145	A.2.1	<i>Attribute Reporting Status</i> Attribute.....	105
146	Annex B	Enhanced Inter-PAN Transmission Mechanism.....	107
147	B.1	Scope and Purpose.....	107
148	B.2	General Description	107
149	B.2.1	What EnhancedInter-PAN Transmission Does	107
150	B.3	Service Specification	108
151	B.3.1	The INTRP-DATA.request Primitive.....	109
152	B.3.1.1	Semantics of the Service Primitive	109
153	B.3.1.2	When Generated.....	111
154	B.3.1.3	Effect on Receipt	111
155	B.3.2	The INTRP-DATA.confirm Primitive	111
156	B.3.2.1	Semantics of the Service Primitive	111
157	B.3.2.2	When Generated.....	112
158	B.3.2.3	Effect on Receipt	112
159	B.3.3	The INTRP-DATA.indication Primitive	112
160	B.3.3.1	Semantics of the Service Primitive	112
161	B.3.3.2	When Generated.....	114
162	B.3.3.3	Effect on Receipt	114
163	B.3.4	Qualifying and Testing of Inter-PAN Messages	114
164	B.4	Frame Formats.....	114
165	B.5	Frame Processing	117
166	B.5.1	Enhanced Inter-PAN Transmission.....	117
167	B.5.2	Enhanced Inter-PAN Reception.....	118
168	B.6	Initiating a Enhanced Inter-PAN Interaction.....	119
169	B.7	Best Practices	120
170	B.8	Security Requirements.....	120
171	Annex C	Key Establishment Cluster.....	121
172	C.1	Scope and Purpose.....	121
173	C.2	General Description	121
174	C.2.1	Introduction	121
175	C.2.2	Security Credentials	121
176	C.2.3	Network Security	121
177	C.2.4	Key Establishment.....	121
178	C.2.5	Symmetric Key Key Establishment	122
179	C.2.6	Public Key Key Establishment.....	123
180	C.2.7	General Exchange	123
181	C.2.7.1	Exchange Static and Ephemeral Data.....	124

182	C.2.7.2	Generate Key Bitstream.....	125
183	C.2.7.3	Derive MAC Key and Key Data	125
184	C.2.7.4	Confirm Key Using MAC	125
185	C.3	Cluster List	126
186	C.3.1	Key Establishment Cluster	126
187	C.3.1.1	Overview	126
188	C.3.1.1.1	Negotiating the Key Establishment Suite	128
189	C.3.1.2	Server.....	128
190	C.3.1.2.1	Dependencies	128
191	C.3.1.2.2	Attributes.....	128
192	C.3.1.2.3	Commands Received.....	129
193	C.3.1.2.4	Commands Generated	134
194	C.3.1.3	Client	135
195	C.3.1.3.1	Dependencies	135
196	C.3.1.3.2	Attributes.....	135
197	C.3.1.3.3	Commands Received.....	136
198	C.3.1.3.4	Commands Generated	141
199	C.4	Application Implementation	141
200	C.4.1	Network Security for Smart Energy Networks.....	141
201	C.4.2	Certificate-Based Key Establishment	142
202	C.4.2.1	Notation and Representation.....	143
203	C.4.2.1.1	Strings and String Operations	143
204	C.4.2.1.2	Integers and their Representation	143
205	C.4.2.1.3	Entities	143
206	C.4.2.2	Cryptographic Suite 1 Building Blocks.....	143
207	C.4.2.2.1	Elliptic-Curve Domain Parameters	143
208	C.4.2.2.2	Elliptic-Curve Point Representation.....	143
209	C.4.2.2.3	Elliptic-Curve Key Pair	144
210	C.4.2.2.4	ECC Implicit Certificates	144
211	C.4.2.2.5	Block-Cipher.....	144
212	C.4.2.2.6	Cryptographic Hash Function	145
213	C.4.2.2.7	Keyed Hash Function for Message Authentication	145
214	C.4.2.2.8	Derived Shared Secret	145
215	C.4.2.3	Cryptographic Suite 2 Building Blocks.....	145
216	C.4.2.3.1	Elliptic-Curve Point Representation.....	145
217	C.4.2.3.2	Elliptic-Curve Key Pair	146
218	C.4.2.3.3	ECC Implicit Certificates	146
219	C.4.2.3.4	Block-Cipher.....	148
220	C.4.2.3.5	Cryptographic Hash Function	148
221	C.4.2.3.6	Keyed Hash Function for Message Authentication	148
222	C.4.2.3.7	Derived Shared Secret	148
223	C.4.2.4	Certificate-Based Key-Establishment.....	148
224	C.4.2.4.1	Exchange Ephemeral Data	150

225	C.4.2.4.2	Validate Implicit Certificates.....	150
226	C.4.2.4.3	Derive Keying Material.....	150
227	C.4.2.4.4	Confirm Keys.....	152
228	C.5	Key Establishment Test Vectors for Cryptographic Suite 1	152
229	C.5.1	Preconfigured Data	152
230	C.5.1.1	CA Public Key.....	152
231	C.5.1.2	Responder Data.....	153
232	C.5.1.3	Initiator Data.....	153
233	C.5.2	Key Establishment Messages	154
234	C.5.2.1	Initiate Key Establishment Request	155
235	C.5.2.2	Initiate Key Establishment Response	156
236	C.5.2.3	Ephemeral Data Request	157
237	C.5.2.4	Ephemeral Data Response	157
238	C.5.2.5	Confirm Key Request	158
239	C.5.2.6	Confirm Key Response.....	159
240	C.5.3	Data Transformation	160
241	C.5.3.1	ECMQV Primitives	160
242	C.5.3.2	Key Derivation Function (KDF)	160
243	C.5.3.3	Initiator Transform.....	161
244	C.5.3.3.1	Ephemeral Data	161
245	C.5.3.3.2	Step Summary	161
246	C.5.3.3.3	Detailed Steps.....	161
247	C.5.3.4	Responder Transform.....	163
248	C.5.3.4.1	Ephemeral Data	163
249	C.5.3.4.2	Step Summary	163
250	C.5.3.4.3	Detailed Steps	163
251	C.6	Key Establishment Test Vectors for Cryptographic Suite 2.....	164
252	C.6.1	Preconfigured Data	165
253	C.6.1.1	CA Public Key.....	165
254	C.6.1.2	Responder Data.....	165
255	C.6.1.3	Initiator Data.....	166
256	C.6.2	Key Establishment Messages	167
257	C.6.2.1	Initiate Key Establishment Request	167
258	C.6.2.2	Initiate Key Establishment Response	168
259	C.6.2.3	Ephemeral Data Request	169
260	C.6.2.4	Ephemeral Data Response	169
261	C.6.2.5	Confirm Key Request	170
262	C.6.2.6	Confirm Key Response.....	171
263	C.6.3	Data Transformation	171
264	C.6.3.1	ECMQV Primitives	172
265	C.6.3.2	Key Derivation Function (KDF)	172
266	C.6.3.3	Initiator Transform.....	172
267	C.6.3.3.1	Ephemeral Data	172

268	C.6.3.3.2 Step Summary	172
269	C.6.3.3.3 Detailed Steps.....	173
270	C.6.3.4 Responder Transform.....	174
271	C.6.3.4.1 Ephemeral Data.....	174
272	C.6.3.4.2 Step Summary	175
273	C.6.3.4.3 Detailed Steps	175
274	Annex D Smart Energy Cluster Descriptions.....	177
275	D.1 Annex Guidelines.....	177
276	D.1.1 Client/Server Model Information.....	177
277	D.1.2 Interpretation of Reserved Field Values or Bitmaps	177
278	D.2 Demand Response and Load Control Cluster	178
279	D.2.1 Overview	178
280	D.2.2 Server	178
281	D.2.2.1 Dependencies	178
282	D.2.2.2 Attributes	178
283	D.2.2.3 Commands Generated.....	179
284	D.2.2.3.1 Load Control Event Command	179
285	D.2.2.3.2 Cancel Load Control Event Command	185
286	D.2.2.3.3 Cancel All Load Control Events Command	186
287	D.2.2.4 Commands Received	187
288	D.2.3 Client	187
289	D.2.3.1 Dependencies	187
290	D.2.3.2 Client Cluster Attributes	188
291	D.2.3.2.1 Utility Enrollment Group Attribute.....	188
292	D.2.3.2.2 Start Randomization Minutes Attribute.....	189
293	D.2.3.2.3 End Randomization Minutes Attribute.....	189
294	D.2.3.2.4 DeviceClassValue Attribute	189
295	D.2.3.3 Commands Generated.....	189
296	D.2.3.3.1 Report Event Status Command.....	190
297	D.2.3.3.2 Get Scheduled Events Command	193
298	D.2.3.4 Commands Received	194
299	D.2.3.5 Attribute Reporting.....	194
300	D.2.4 Application Guidelines	194
301	D.2.4.1 Load Control Rules, Server.....	195
302	D.2.4.1.1 Load Control Server, Identifying Use of SetPoint and Offset Fields	195
303	D.2.4.1.2 Load Control Server, Editing of Scheduled Events	195
304	D.2.4.2 Load Control Rules, Client	195
305	D.2.4.2.1 Start and Stop Randomization	195
306	D.2.4.2.2 Editing of DR Control Parameters.....	196
307	D.2.4.2.3 Response to Price Events + Load Control Events	196
308	D.2.4.2.4 Opt-Out Messages	196
309	D.2.4.2.5 Thermostat/HVAC Controls	196
310	D.2.4.2.6 Demand Response and Load Control Transaction Examples	196

311	D.3 Metering Cluster	199
312	D.3.1 Overview	199
313	D.3.2 Server	202
314	D.3.2.1 Dependencies.....	202
315	D.3.2.2 Attributes	202
316	D.3.2.2.1 Reading Information Set	202
317	D.3.2.2.2 Summation TOU Information Set	212
318	D.3.2.2.3 Meter Status Attribute Set	215
319	D.3.2.2.4 Formatting.....	223
320	D.3.2.2.5 Historical Consumption Attribute	235
321	D.3.2.2.6 Load Profile Configuration	245
322	D.3.2.2.7 Supply Limit Attributes.....	246
323	D.3.2.2.8 Block Information Attribute Set (Delivered)	249
324	D.3.2.2.9 Alarms Set.....	253
325	D.3.2.2.10 Block Information Attribute Set (Received).....	259
326	D.3.2.2.11 Meter Billing Attribute Set	263
327	D.3.2.2.12 Supply Control Attribute Set	265
328	D.3.2.2.13 Alternative Historical Consumption Attribute Set	266
329	D.3.2.3 Server Commands.....	275
330	D.3.2.3.1 Commands Generated	275
331	D.3.3 Client	301
332	D.3.3.1 Dependencies.....	301
333	D.3.3.2 Attributes	301
334	D.3.3.2.1 Notification Attribute Set.....	302
335	D.3.3.3 Client Commands	306
336	D.3.3.3.1 Commands Generated	306
337	D.3.4 Metering Application Guidelines.....	323
338	D.3.4.1 Attribute Reporting.....	323
339	D.3.4.2 Fast Polling or Reporting for Monitoring Energy Savings	323
340	D.3.4.3 Metering Data Updates.....	323
341	D.3.4.3.1 Fast Polling Periods	323
342	D.3.4.4 Mirroring	323
343	D.3.4.4.1 Discovery	324
344	D.3.4.4.2 Mirror Attributes	324
345	D.3.4.4.3 Two Way Mirror for BOMD	324
346	D.3.4.5 An Introduction to Snapshots	331
347	D.3.4.6 Supply Control.....	331
348	D.4 Price Cluster	332
349	D.4.1 Overview	332
350	D.4.2 Server	332
351	D.4.2.1 Dependencies.....	332
352	D.4.2.2 Attributes	332
353	D.4.2.2.1 Tier Label (Delivered) Set.....	333

354	D.4.2.2.2	Block Threshold (Delivered) Set.....	335
355	D.4.2.2.3	Block Period (Delivered) Set	338
356	D.4.2.2.4	Commodity Set.....	339
357	D.4.2.2.5	Block Price Information (Delivered) Set.....	341
358	D.4.2.2.6	Extended Price Information (Delivered) Set.....	346
359	D.4.2.2.7	Tariff Information (Delivered) Attribute Set	346
360	D.4.2.2.8	Billing Information (Delivered) Attribute Set	350
361	D.4.2.2.9	Credit Payment Attribute Set	352
362	D.4.2.2.10	Received Tier Label Attribute Set	354
363	D.4.2.2.11	Received Block Threshold Attribute Set	355
364	D.4.2.2.12	Received Block Period Attribute Set	355
365	D.4.2.2.13	Received Block Price Information Attribute Set	356
366	D.4.2.2.14	Received Extended Price Information Attribute Set.....	357
367	D.4.2.2.15	Received Tariff Information Attribute Set.....	357
368	D.4.2.2.16	Received Billing Information Attribute Set.....	359
369	D.4.2.3	Commands Received	360
370	D.4.2.3.1	Error Handling	361
371	D.4.2.3.2	GetCurrentPrice Command.....	361
372	D.4.2.3.3	GetScheduledPrices Command	362
373	D.4.2.3.4	PriceAcknowledgement Command	363
374	D.4.2.3.5	GetBlockPeriod(s) Command	363
375	D.4.2.3.6	GetConversionFactor Command.....	364
376	D.4.2.3.7	GetCalorificValue Command	365
377	D.4.2.3.8	GetTariffInformation Command.....	366
378	D.4.2.3.9	GetPriceMatrix Command	367
379	D.4.2.3.10	GetBlockThresholds Command.....	367
380	D.4.2.3.11	GetCO ₂ Value Command.....	368
381	D.4.2.3.12	GetTierLabels Command	369
382	D.4.2.3.13	GetBillingPeriod Command	369
383	D.4.2.3.14	GetConsolidatedBill Command	370
384	D.4.2.3.15	CPPEventResponse Command	371
385	D.4.2.3.16	GetCreditPayment Command	372
386	D.4.2.3.17	GetCurrencyConversion Command	372
387	D.4.2.3.18	GetTariffCancellation Command.....	373
388	D.4.2.4	Commands Generated.....	373
389	D.4.2.4.1	Publish Price Command.....	374
390	D.4.2.4.2	Publish Block Period Command	384
391	D.4.2.4.3	PublishConversionFactor Command.....	388
392	D.4.2.4.4	PublishCalorificValue Command	389
393	D.4.2.4.5	PublishTariffInformation Command	389
394	D.4.2.4.6	PublishPriceMatrix Command	392
395	D.4.2.4.7	PublishBlockThresholds Command.....	394
396	D.4.2.4.8	PublishCO ₂ Value Command	397

397	D.4.2.4.9	PublishTierLabels Command.....	398
398	D.4.2.4.10	PublishBillingPeriod Command	400
399	D.4.2.4.11	PublishConsolidatedBill Command.....	401
400	D.4.2.4.12	PublishCPPEvent Command	403
401	D.4.2.4.13	PublishCreditPayment Command.....	405
402	D.4.2.4.14	PublishCurrencyConversion Command.....	407
403	D.4.2.4.15	CancelTariff Command	408
404	D.4.3	Client	409
405	D.4.3.1	Dependencies.....	409
406	D.4.3.2	Attributes	409
407	D.4.3.2.1	PriceIncreaseRandomizeMinutes Attribute	410
408	D.4.3.2.2	PriceDecreaseRandomizeMinutes Attribute.....	410
409	D.4.3.2.3	CommodityType Attribute	411
410	D.4.3.3	Commands Received	411
411	D.4.3.4	Commands Generated.....	411
412	D.4.4	Application Guidelines	411
413	D.4.4.1	Registering for Commands	411
414	D.4.4.2	Attribute Reporting.....	411
415	D.4.4.3	Block Tariffs.....	411
416	D.4.4.3.1	TOU Charging Only	412
417	D.4.4.3.2	Block Charging only	412
418	D.4.4.3.3	Block/TOU Combination Charging	412
419	D.4.4.3.4	Application Guidelines for Block Pricing under specific events	413
420	D.4.4.4	Handling of Enhanced Tariffs	413
421	D.4.4.4.1	Block Charging	414
422	D.4.4.4.2	TOU Charging	415
423	D.4.4.4.3	TOU/Block Charging.....	415
424	D.4.4.4.4	Critical Peak Pricing	415
425	D.4.4.4.5	Generation Charging	416
426	D.5	Messaging Cluster.....	417
427	D.5.1	Overview	417
428	D.5.2	Server	417
429	D.5.2.1	Dependencies.....	417
430	D.5.2.2	Attributes	417
431	D.5.2.3	Commands Generated.....	417
432	D.5.2.3.1	Display Message Command.....	418
433	D.5.2.3.2	Cancel Message Command	421
434	D.5.2.3.3	Display Protected Message Command.....	421
435	D.5.2.3.4	Cancel All Messages Command	421
436	D.5.3	Client	422
437	D.5.3.1	Dependencies.....	422
438	D.5.3.2	Attributes	422
439	D.5.3.3	Commands Generated.....	422

440	D.5.3.3.1	Get Last Message Command.....	422
441	D.5.3.3.2	Message Confirmation Command.....	423
442	D.5.3.3.3	GetMessageCancellation Command.....	424
443	D.5.4	Application Guidelines	424
444	D.6	Tunneling Cluster	426
445	D.6.1	Overview	426
446	D.6.2	Server	429
447	D.6.2.1	Dependencies.....	429
448	D.6.2.2	Attributes	429
449	D.6.2.2.1	CloseTunnelTimeout Attribute.....	430
450	D.6.2.3	Parameters	430
451	D.6.2.4	Commands Received	430
452	D.6.2.4.1	RequestTunnel Command	431
453	D.6.2.4.2	CloseTunnel Command.....	432
454	D.6.2.4.3	TransferData Command	433
455	D.6.2.4.4	TransferDataError Command.....	434
456	D.6.2.4.5	AckTransferData Command.....	435
457	D.6.2.4.6	ReadyData Command	436
458	D.6.2.4.7	Get Supported Tunnel Protocols Command	437
459	D.6.2.5	Commands Generated.....	437
460	D.6.2.5.1	RequestTunnelResponse Command.....	438
461	D.6.2.5.2	TransferData Command	439
462	D.6.2.5.3	TransferDataError Command.....	440
463	D.6.2.5.4	AckTransferData Command.....	440
464	D.6.2.5.5	ReadyData Command	440
465	D.6.2.5.6	Supported Tunnel Protocols Response Command	440
466	D.6.2.5.7	TunnelClosureNotification Command.....	441
467	D.6.3	Client	442
468	D.6.3.1	Dependencies.....	442
469	D.6.3.2	Attributes	442
470	D.6.3.3	Commands Received	442
471	D.6.3.4	Commands Generated.....	442
472	D.7	Prepayment Cluster	443
473	D.7.1	Overview	443
474	D.7.2	Server	444
475	D.7.2.1	Dependencies.....	444
476	D.7.2.2	Attributes	444
477	D.7.2.2.1	Prepayment Information Attribute Set	445
478	D.7.2.2.2	Top-up Attribute Set	452
479	D.7.2.2.3	Debt Attribute Set	455
480	D.7.2.2.4	Supply Control Set	458
481	D.7.2.2.5	Alarms Attribute Set	458
482	D.7.2.2.6	Historical Cost Consumption Information Set.....	461

483	D.7.2.3 Commands Received	471
484	D.7.2.3.1 Select Available Emergency Credit Command	471
485	D.7.2.3.2 Change Supply Command.....	472
486	D.7.2.3.3 Change Debt Command	472
487	D.7.2.3.4 Emergency Credit Setup Command.....	474
488	D.7.2.3.5 Consumer Top Up Command	475
489	D.7.2.3.6 Credit Adjustment Command	476
490	D.7.2.3.7 Change Payment Mode Command	477
491	D.7.2.3.8 Get Prepay Snapshot Command	479
492	D.7.2.3.9 Get Top Up Log	480
493	D.7.2.3.10 Set Low Credit Warning Level	480
494	D.7.2.3.11 Get Debt Repayment Log Command.....	481
495	D.7.2.3.12 Set Maximum Credit Limit.....	481
496	D.7.2.3.13 Set Overall Debt Cap	483
497	D.7.2.4 Commands Generated.....	483
498	D.7.2.4.1 Supply Status Response Command	484
499	D.7.2.4.2 Publish Prepay Snapshot Command	484
500	D.7.2.4.3 Change Payment Mode Response Command.....	487
501	D.7.2.4.4 Consumer Top Up Response Command.....	488
502	D.7.2.4.5 Publish Top Up Log Command	490
503	D.7.2.4.6 Publish Debt Log Command.....	491
504	D.7.3 Client	492
505	D.7.3.1 Dependencies	492
506	D.7.3.2 Attributes	492
507	D.7.3.3 Commands Received	492
508	D.7.3.4 Commands Generated.....	492
509	D.7.4 Application Guidelines	492
510	D.7.4.1 Credit Status Attribute	492
511	D.7.4.1.1 Statuses Explained - an Example	494
512	D.8 Over-the-Air Bootload Cluster.....	496
513	D.8.1 Overview	496
514	D.8.2 OTA Bootloading Timing Considerations	496
515	D.9 Calendar Cluster	497
516	D.9.1 Overview	497
517	D.9.2 Server	500
518	D.9.2.1 Dependencies	500
519	D.9.2.2 Attributes	500
520	D.9.2.2.1 Auxiliary Switch Label Attribute Set	500
521	D.9.2.3 Commands Generated.....	501
522	D.9.2.3.1 PublishCalendar Command	501
523	D.9.2.3.2 PublishDayProfile Command	503
524	D.9.2.3.3 PublishWeekProfile Command.....	507
525	D.9.2.3.4 PublishSeasons Command	508

526	D.9.2.3.5 PublishSpecialDays Command	509
527	D.9.2.3.6 CancelCalendar Command	512
528	D.9.2.4 Commands Received	513
529	D.9.2.4.1 GetCalendar Command.....	513
530	D.9.2.4.2 GetDayProfiles Command	514
531	D.9.2.4.3 GetWeekProfiles Command	515
532	D.9.2.4.4 GetSeasons Command	516
533	D.9.2.4.5 GetSpecialDays Command	516
534	D.9.2.4.6 GetCalendarCancellation Command	517
535	D.9.3 Client	517
536	D.9.3.1 Dependencies	517
537	D.9.3.2 Attributes	518
538	D.9.3.3 Commands Received	518
539	D.9.3.4 Commands Generated.....	518
540	D.10 Device Management Cluster.....	519
541	D.10.1 Overview	519
542	D.10.1.1 Supplier Control	519
543	D.10.1.2 Tenancy Control	519
544	D.10.1.3 Password Control.....	519
545	D.10.2 Server.....	520
546	D.10.2.1 Dependencies	520
547	D.10.2.2 Attributes	520
548	D.10.2.2.1 Supplier Control Attribute Set	520
549	D.10.2.2.2 Tenancy Control Attribute Set	524
550	D.10.2.2.3 Backhaul Control Attribute Set.....	525
551	D.10.2.2.4 HAN Control Attribute Set	525
552	D.10.2.3 Commands Received	526
553	D.10.2.3.1 Get Change of Tenancy Command.....	526
554	D.10.2.3.2 Get Change of Supplier Command	526
555	D.10.2.3.3 RequestNewPassword Command	527
556	D.10.2.3.4 GetSiteID Command.....	527
557	D.10.2.3.5 Report Event Configuration Command	527
558	D.10.2.3.6 GetCIN Command	528
559	D.10.2.4 Commands Generated.....	529
560	D.10.2.4.1 Publish Change of Tenancy Command.....	529
561	D.10.2.4.2 Publish Change of Supplier Command	530
562	D.10.2.4.3 Request New Password Response Command.....	532
563	D.10.2.4.4 Update SiteID Command.....	533
564	D.10.2.4.5 SetEventConfiguration Command	534
565	D.10.2.4.6 GetEventConfiguration Command	536
566	D.10.2.4.7 Update CIN Command	537
567	D.10.3 Client	538
568	D.10.3.1 Dependencies.....	538

569	D.10.3.2 Attributes	538
570	D.10.3.2.1 Supplier Attribute Set	538
571	D.10.3.2.2 Price Event Configuration Attribute Set	539
572	D.10.3.2.3 Metering Event Configuration Attribute Set.....	543
573	D.10.3.2.4 Messaging Event Configuration Attribute Set.....	552
574	D.10.3.2.5 Prepayment Event Configuration Attribute Set	553
575	D.10.3.2.6 Calendar Event Configuration Attribute Set.....	557
576	D.10.3.2.7 Device Management Event Configuration Attribute Set	558
577	D.10.3.2.8 Tunnel Event Configuration Attribute Set.....	561
578	D.10.3.2.9 OTA Event Configuration Attribute Set	562
579	D.10.3.3 Commands Received	564
580	D.10.3.4 Commands Generated.....	564
581	D.10.4 Application Guidelines	564
582	D.10.4.1 Passwords	564
583	D.10.4.2 Consumer Password Use Case	564
584	D.10.4.3 Engineer Password Use Case	564
585	D.10.4.4 Password Security Recommendations.....	565
586	D.11 Events Cluster	566
587	D.11.1 Overview	566
588	D.11.2 Server.....	567
589	D.11.2.1 Dependencies.....	567
590	D.11.2.2 Attributes	567
591	D.11.2.3 Commands Received	567
592	D.11.2.3.1 Get Event Log Command	568
593	D.11.2.3.2 Clear Event Log Request Command.....	570
594	D.11.2.4 Commands Generated.....	570
595	D.11.2.4.1 Publish Event Command.....	571
596	D.11.2.4.2 Publish Event Log Command	572
597	D.11.2.4.3 Clear Event Log Response Command	573
598	D.11.3 Client	574
599	D.11.3.1 Dependencies.....	574
600	D.11.3.2 Attributes	574
601	D.11.3.3 Commands Received	575
602	D.11.3.4 Commands Generated.....	575
603	D.12 Energy Management Cluster	576
604	D.12.1 Overview	576
605	D.12.2 Server.....	576
606	D.12.2.1 Dependencies.....	576
607	D.12.2.2 Attributes	576
608	D.12.2.2.1 Load Control State Attribute.....	577
609	D.12.2.2.2 CurrentEventID Attribute	578
610	D.12.2.2.3 CurrentEventStatus Attribute	578
611	D.12.2.2.4 Conformance Level Attribute	580

612	D.12.2.2.5 MinimumOffTime, MinimumONTime & MinimumCyclePeriod Attributes	580
613	D.12.2.3 Attribute Reporting.....	581
614	D.12.2.4 Commands Received	581
615	D.12.2.4.1 Manage Event Command.....	581
616	D.12.2.5 Commands Generated.....	582
617	D.12.2.5.1 Report Event Status Command.....	583
618	D.12.3 Client	584
619	D.12.3.1 Dependencies.....	584
620	D.12.3.2 Attributes	584
621	D.12.3.3 Commands Received	584
622	D.12.3.4 Commands Generated.....	584
623	D.13 MDU Pairing Cluster	585
624	D.13.1 Overview	585
625	D.13.2 Server.....	585
626	D.13.2.1 Dependencies.....	585
627	D.13.2.2 Attributes	585
628	D.13.2.3 Commands Generated.....	585
629	D.13.2.3.1 Pairing Response Command	586
630	D.13.3 Client	587
631	D.13.3.1 Dependencies.....	587
632	D.13.3.2 Attributes	587
633	D.13.3.3 Commands Generated.....	587
634	D.13.3.3.1 Pairing Request Command	587
635	Annex E Rules And Guidelines For Overlapping Events	589
636	E.1 Definitions	589
637	E.2 Rules and Guideline.....	590
638	E.3 Event Examples.....	591
639	E.3.1 Correct Overlapping Events for Different Device Classes	592
640	E.3.2 Correct Superseded Event for a Device Class	593
641	E.3.3 Superseding Events for Subsets of Device Classes.....	593
642	E.3.4 Ending Randomization Between Events	594
643	E.3.5 Start Randomization Between Events	595
644	E.3.6 Acceptable Gaps Caused by Start and Stop Randomization of Events	596
645	Annex F Joining Procedure Using Pre-Configured Trust Center Link Keys	598

List of Tables

646	Table 1-1 – Document Revision Change History	31
647	Table 5-1 – Startup Parameters	45
648	Table 5-2 – Join Parameters	47
649	Table 5-3 – Security Parameters	48
650	Table 5-4 – End Device Parameters	49
651	Table 5-5 – Link Status Parameters	49
652	Table 5-6 – Concentrator Parameters.....	49
653	Table 5-7 – APS Transport Parameters	50
654	Table 5-8 – APS Fragmentation Parameters	50
655	Table 5-9 – Binding Parameters	51
656	Table 5-10 – Per SE Network Storage Requirements	57
657	Table 5-11 – Example Hash of the TC Link Key	59
658	Table 5-12 – Parameters of Trust Center Swap-Out	59
659	Table 5-13 – Security Key Assignments per Cluster	63
660	Table 5-14 – Devices Specified in the Smart Energy Profile	85
661	Table 5-15 – Clusters Used in the Smart Energy Profile	87
662	Table 6-1 – Clusters Common to All Devices	92
663	Table 6-2 – Common Features and Functions Configuration for a Smart Energy Device	93
664	Table 6-3 – Clusters Supported by the Energy Service Interface	95
665	Table 6-4 – Clusters Supported by the Metering Device	96
666	Table 6-5 – Clusters Supported by the In-Home Display Device.....	98
667	Table 6-6 – Clusters Supported by the PCT	99
668	Table 6-7 – Clusters Supported by the Load Control Device	100
669	Table 6-8 – Clusters Supported by the Smart Appliance Device.....	101
670	Table 6-9 – Clusters Supported by the Prepayment Terminal Device	102
671	Table 6-10 – Clusters Supported by the Remote Communications Device	103
672	Table A-1– Additional Status Enumeration	105
673	Table A-2– <i>Attribute Reporting Status</i> Attribute	105
674	Table A-3– Attribute Reporting Status Enumerations	105
675	Table B-1– Parameters of the INTRP-DATA. request	110
676	Table B-2– Parameters of the INTRP-DATA. confirm	111
677	Table B-3– Parameters of the INTRP-DATA.indication.....	113
678	Table C-1– Clusters Specified for the Secure Communication Functional Domain	126
679	Table C-2– Key Establishment Attribute Sets	128
680	Table C-3– Key Establishment Attribute Sets	129
681	Table C-4– Values of the <i>KeyEstablishmentSuite</i> Attribute	129
682	Table C-5– Received Command IDs for the Key Establishment Cluster Server.....	129
683	Table C-6– <i>Terminate Key Establishment</i> Command Status Field.....	133
684	Table C-7– Key Establishment Attribute Sets	135
685	Table C-8– Attributes of the Information Attribute Set.....	135
686	Table C-9– Values of the <i>KeyEstablishmentSuite</i> Attribute	136
687	Table C-9– Values of the <i>KeyEstablishmentSuite</i> Attribute	136

688	Table C-10– Received Command IDs for the Key Establishment Cluster Client	136
689	Table C-11– <i>Terminate Key Establishment</i> Command Status Field.....	140
690	Table C-12– Values of the <i>KeyUsage</i> Field.....	147
691	Table C-12– ECC Implicit Certificate format.....	147
692	Table C-13– Parameters Used by Methods of the CBKE Protocol	149
693	Table D-1– Command IDs for the Demand Response and Load Control Server	179
694	Table D-2– Device Class Field BitMap/Encoding	180
695	Table D-3– Criticality Levels.....	181
696	Table D-4– Event Control Field BitMap.....	184
697	Table D-5– Cancel Control	186
698	Table D-6– Cancel All Command Cancel Control Field	187
699	Table D-7– Demand Response Client Cluster Attributes.....	188
700	Table D-8– Generated Command IDs for the Demand Response and Load Control Client.....	190
701	Table D-9– Event Status Field Values	191
702	Table D-10– Metering Cluster Server Attribute Sets.....	202
703	Table D-11– Reading Information Attribute Set.....	203
704	Table D-12– Block Enumerations	207
705	Table D-13– Supply Status Attribute Enumerations	209
706	Table D-14– TOU Information Attribute Set.....	212
707	Table D-15– Meter Status Attribute Set.....	216
708	Table D-16– Mapping of the <i>Status</i> Attribute (Electricity).....	216
709	Table D-17– Meter Status Attribute (Gas)	217
710	Table D-18– Meter Status Attribute (Water)	218
711	Table D-19– Meter Status Attribute (Heat and Cooling).....	218
712	Table D-20– General Flags of the Extended Status BitMap	220
713	Table D-21– Electricity -Meter specific Flags of the Extended Status BitMap.....	221
714	Table D-22– Gas-Meter specific Flags of the Extended Status BitMap	222
715	Table D-23– LowMediumHighStatus Attribute	223
716	Table D-24– Formatting Examples	224
717	Table D-25– Formatting Attribute Set	224
718	Table D-26– <i>UnitofMeasure</i> Attribute Enumerations	226
719	Table D-27– <i>MeteringDeviceType</i> Attribute	229
720	Table D-28– <i>TemperatureUnitOfMeasure</i> Enumeration	232
721	Table D-29– Historical Attribute Set	235
722	Table D-30– Load Profile Configuration Attribute Set	245
723	Table D-31– Supply Limit Attribute Set.....	246
724	Table D-32– Block Information Attribute Set (Delivered)	249
725	Table D-33– Alarm Attribute Set	253
726	Table D-34– Alarm Code Groups	254
727	Table D-35– Generic Alarm Group.....	255
728	Table D-36– Electricity Alarm Group.....	256
729	Table D-37– Generic Flow/Pressure Alarm Group	256

730	Table D-38– Water Specific Alarm Group.....	257
731	Table D-39– Heat and Cooling Specific Alarm Group.....	257
732	Table D-40– Gas Specific Alarm Group	257
733	Table D-41– Extended Generic Alarm Group.....	257
734	Table D-42– Manufacturer Specific Alarm Group	259
735	Table D-43– Block Information Attribute Set (Received)	259
736	Table D-44– Meter Billing Attribute Set	263
737	Table D-45– Supply Control Attribute Set	265
738	Table D-46– Alternative Historical Consumption Attribute Set.....	267
739	Table D-46– Generated Command IDs for the Metering Server.....	275
740	Table D-47– Status Field Values	277
741	Table D-48– ProfileIntervalPeriod Timeframes.....	277
742	Table D-49– Snapshot Schedule Confirmation.....	280
743	Table D-50– Snapshot Confirmation	281
744	Table D-51– Snapshot Cause BitMap.....	282
745	Table D-52– Snapshot Payload Type.....	283
746	Table D-53– Sample Type Enumerations	292
747	Table D-54– Notification Flags Order	296
748	Table D-55– Supply Status Field Enumerations	300
749	Table D-56 – Metering Cluster Client Attribute Sets	301
750	Table D-57 – Notification Attribute Set.....	302
751	Table D-58 – Functional Notification Flags	303
752	Table D-59 – Example Usage of Stay Awake Request Flags	304
753	Table D-60 – Push Historical Metering Data Definition	305
754	Table D-61 – Push Historical Payment Data Attribute Definition.....	305
755	Table D-63– Generated Command IDs for the Metering Client.....	307
756	Table D-64– Interval Channel Values.....	308
757	Table D-65– Snapshot Schedule BitMap	311
758	Table D-66– Supply Control Bits.....	318
759	Table D-67– Local Change Supply: Supply Status Field Enumerations	319
760	Table D-68– SetSupplyStatus: Field Enumerations	321
761	Table D-68– Notification Flags 2.....	328
762	Table D-69– Notification Flags 3.....	329
763	Table D-70– Notification Flags 4.....	329
764	Table D-71– Notification Flags 5.....	330
765	Table D-72– Price Cluster Attribute Sets	333
766	Table D-73– Tier Label Attribute Set	333
767	Table D-74– Block Threshold Attribute Set	335
768	Table D-75– Block Period Attribute Set	338
769	Table D-76– <i>Commodity</i> Attribute Set	340
770	Table D-77– Values and Descriptions for the <i>CalorificValueUnit</i> Attribute	341
771	Table D-78– Block Price Information Attribute Set	342
772	Table D-79– Extended Price Information Set (TOU charging only)	346

773	Table D-80– Tariff Information Attribute Set.....	346
774	Table D-81– TierBlockMode Enumeration	348
775	Table D-82– CO ₂ Unit Enumeration.....	350
776	Table D-83– Billing Information Attribute Set.....	351
777	Table D-84– Credit Payment Attribute Set	352
778	Table D-85– CreditPaymentStatus Enumeration	353
779	Table D-86– PaymentDiscountDuration Enumerations.....	353
780	Table D-87– Received Tier Label Attribute Set	354
781	Table D-88– Received Block Threshold Attribute Set	355
782	Table D-89– Received Block Period Attribute Set	355
783	Table D-90– Received Block Price Attribute Set	356
784	Table D-91– Received Extended Price Information Attribute Set (TOU charging only).....	357
785	Table D-92– Received Tariff Information Attribute Set.....	357
786	Table D-93– Received Billing Information Attribute Set	359
787	Table D-94– Received Command IDs for the Price Cluster	360
788	Table D-95– Generated Command IDs for the Price Cluster.....	373
789	Table D-96– Price Tier Sub-field Enumerations	376
790	Table D-97– Register Tier Sub-field Enumerations.....	378
791	Table D-98– Alternate Cost Unit Enumerations	379
792	Table D-99– Price Control Field BitMap	380
793	Table D-100– Generation Tier Enumerations	380
794	Table D-101– Extended Price Tier Field Enumerations	382
795	Table D-102– Extended Register Tier Field Enumerations	383
796	Table D-103– Block Period Control Field BitMap	386
797	Table D-104– Block Period DurationTimebase Enumeration	386
798	Table D-105– Block Period Duration Control Enumeration	387
799	Table D-106– Tariff Resolution Period Enumeration	387
800	Table D-107– Tariff Type Enumeration	391
801	Table D-108– Tariff Charging Scheme Enumeration	391
802	Table D-109– PublishPriceMatrix Sub-Payload Control Bitmap	394
803	Table D-110– PublishBlockThresholds Sub-Payload Control Bitmap.....	396
804	Table D-111– CPP Auth Enumeration	404
805	Table D-112– Currency Change Control	408
806	Table D-113– Price Client Cluster Attributes	409
807	Table D-114– Generated Command IDs for the Messaging Server.....	417
808	Table D-115– Message Control Field Bit Map	418
809	Table D-116– Extended Message Control Field Bit Map	420
810	Table D-117– Messaging Client Commands	422
811	Table D-118– Message Confirmation Control.....	423
812	Table D-119– Tunneling Server Attributes.....	429
813	Table D-120– Cluster Parameters Passed Through Commands	430
814	Table D-121– Cluster -specific Commands Received by the Server	431
815	Table D-122– ProtocolID Enumerations.....	431

816	Table D-123– TransferDataStatus Values	435
817	Table D-124– Cluster-Specific Commands Sent by the Server	437
818	Table D-125– TunnelStatus Values	438
819	Table D-126– Prepayment Attribute Sets.....	444
820	Table D-127– <i>Prepayment Information</i> Attribute Set.....	445
821	Table D-128– <i>Payment Control Configuration</i> Attribute	447
822	Table D-129– <i>Credit Status</i> Attribute	449
823	Table D-130– OverallDebtCap Example	450
824	Table D-131– Top-up Attribute Set	452
825	Table D-132– Debt Attribute Set	455
826	Table D-133– Debt Recovery Method Enumerations.....	457
827	Table D-134– <i>Recovery Frequency</i> Field Enumerations	457
828	Table D-135– Alarm Attribute Set.....	458
829	Table D-136– Prepayment Alarm Status Indicators	459
830	Table D-137– Alarms Code Group	460
831	Table D-138– PrepayGenericAlarmGroup	460
832	Table D-139– PrepaySwitchAlarmGroup	460
833	Table D-140– PrepayEventAlarmGroup.....	461
834	Table D-141– Historical Cost Consumption Information Attribute Set	461
835	Table D-143– <i>CurrencyScalingFactor</i> Enumerations.....	468
836	Table D-142– Cluster -specific Commands Received by the Server	471
837	Table D-143– Originating Device Field Enumerations	472
838	Table D-144– Debt Amount Type Field Enumerations	473
839	Table D-145– Credit Type Field Enumerations	477
840	Table D-146– Debt Type Field Enumerations	481
841	Table D-147– Cluster -specific Commands Sent by the Server.....	483
842	Table D-148– Snapshot Payload Cause	485
843	Table D-149– Snapshot Payload Type.....	486
844	Table D-150– Friendly Credit BitMap.....	487
845	Table D-151– Result Type Field Enumerations.....	489
846	Table D-152–Calendar Data Structures	498
847	Table D-153–Calendar Cluster Attribute Sets	500
848	Table D-154– Auxiliary Switch Label Attribute Set	500
849	Table D-155– Commands Generated by the Calendar Cluster Server.....	501
850	Table D-156– Calendar Type Enumeration	502
851	Table D-159– Calendar Time Reference Enumeration.....	503
852	Table D-157– Commands Received by the Calendar Cluster Server	513
853	Table D-159– Device Management Attribute Sets	520
854	Table D-160– Supplier Control Attribute Set	520
855	Table D-160– Proposed Change Control	522
856	Table D-161– Contactor State Bit Combinations	523
857	Table D-162– Tenancy Control Attribute Set.....	524
858	Table D-163– Backhaul Control Attribute Set.....	525

859	Table D-164– State of the WAN Connection	525
860	Table D-165– HAN Control Attribute Set	525
861	Table D-166– Cluster Specific Commands Received by the Server	526
862	Table D-167– Cluster Specific Commands Generated by the Server	529
863	Table D-168– Password Type Enumeration	533
864	Table D-169– Configuration Control Enumeration	535
865	Table D-170– Device Management Client Attribute Sets	538
866	Table D-171– Supplier Attribute Sets	538
867	Table D-172– Price Event Configuration Attribute Set	539
868	Table D-173– Event Configuration Bitmaps	542
869	Table D-174– Metering Event Configuration Attribute Set	543
870	Table D-175– Messaging Event Configuration Attribute Set	552
871	Table D-176– Prepayment Event Configuration Attribute Set	553
872	Table D-177– Calendar Event Configuration Attribute Set	557
873	Table D-178– Device Management Event Configuration Attribute Set	558
874	Table D-179– Tunneling Event Configuration Attribute Set	561
875	Table D-180– OTA Event Configuration Attribute Set	563
876	Table D-181– Cluster Specific Commands Received by the Server	567
877	Table D-182– Log ID Enumeration	568
878	Table D-183– Event Control Bitmap	569
879	Table D-184– Cluster Specific Commands Generated by the Server	570
880	Table D-185– Event Action Control Bitmap	571
881	Table D-186– Log Payload Control Bitmap	573
882	Table D-187– ClearedEventsLogs Bitmap	574
883	Table D-187– Energy Management Server Cluster Attributes	576
884	Table D-188– Load Control State BitMap/Encoding	577
885	Table D-189– Current Event Status BitMap/Encoding	578
886	Table D-190– Received Command IDs for the Energy Management Server	581
887	Table D-191– Action(s) Required BitMap/Encoding	582
888	Table D-192– Generated Command IDs for the Energy Management Server	583
889	Table D-193– MDU Pairing Server Commands	585
890	Table D-194– MDU Pairing Client Commands	587
891		

List of Figures

892	List of Figures	
893	Figure 5-1 Utility Private HAN.....	41
894	Figure 5-2 Utility Private NAN.....	42
895	Figure 5-3 Customer Private NAN.....	42
896	Figure 5-4 An Example of a MDU Setup	43
897	Figure 5-5 Successful Join and Registration	62
898	Figure 5-6 Node Communication with Other Nodes on the Network Using APS Layer Encryption	67
899	Figure 5-7 Smart Energy Device Installation Code Process	68
900	Figure 5-8 Installation Code Use with the Trust Center	69
902	Figure B-1– ZigBee Stack with Stub APS	108
903	Figure B-2– Normal ZigBee Frame	114
904	Figure B-3– Enhanced Inter-PAN ZigBee Frame	115
905	Figure B-4– Stub NWK Header Format	115
906	Figure B-5– NWK Frame Control Field	115
907	Figure B-6– Stub APS Header Format	115
908	Figure B-7– APS Frame Control Field.....	116
909	Figure C-1– Overview of General Exchange	124
910	Figure C-2– Typical Usage of the Key Establishment Cluster	126
911	Figure C-3– Key Establishment Command Exchange.....	127
912	Figure C-4– <i>Initiate Key Establishment Request</i> Command Payload.....	130
913	Figure C-5– <i>Ephemeral Data Request</i> Command Payload.....	132
914	Figure C-6– <i>Confirm Key Request</i> Command Payload.....	132
915	Figure C-7– <i>Terminate Key Establishment</i> Command Payload.....	133
916	Figure C-8– <i>Initiate Key Establishment Response</i>	137
917	Figure C-9– <i>Ephemeral Data Response</i> Command Payload	138
918	Figure C-10– <i>Confirm Key Response</i> Command Payload	139
919	Figure C-11– <i>Terminate Key Establishment</i> Command Payload.....	140
920	Figure C-12– Key Establishment Command Exchange.....	155
921	Figure C-13– Key Establishment Command Exchange.....	167
922	Figure D-1– Demand Response/Load Control Cluster Client Server Example.....	178
923	Figure D-2– <i>Load Control Event</i> Command Payload.....	179
924	Figure D-3– <i>Cancel Load Control Event</i> Payload	185
925	Figure D-4– <i>Cancel All Load Control Events</i> Command Payload	187
926	Figure D-5– <i>Report Event Status</i> Command Payload	190
927	Figure D-6– <i>Get Scheduled Events</i> Command Payload	193
928	Figure D-7– Example of Both a Successful and an Overridden Load.....	197
929	Figure D-8– Example of a Load Curtailment Superseded and.....	198
930	Figure D-9– Standalone ESI Model with Mains Powered Metering Device	199
931	Figure D-10– Standalone ESI Model with Battery Powered Metering Device	200
932	Figure D-11– ESI Model with Integrated Metering Device.....	201
933	Figure D-12– <i>Get Profile Response</i> Command Payload	276

934	Figure D-13– <i>Request Fast Poll Mode Response</i> Command Payload.....	279
935	Figure D-14– <i>ScheduleSnapshotResponse</i> Command Payload	280
936	Figure D-15– Snapshot Response Payload Sub-Payload.....	280
937	Figure D-16– <i>TakeSnapshotResponse</i> Command Payload	281
938	Figure D-17– <i>Publish Snapshot</i> Command Payload.....	281
939	Figure D-18– TOU Information Delivered Snapshot Payload.....	284
940	Figure D-19– TOU Information Received Snapshot Payload	285
941	Figure D-20– Block Information Delivered SnapshotPayload	286
942	Figure D-21– Block Information Received SnapshotPayload	287
943	Figure D-22– TOU Information Delivered (No Billing) Snapshot Payload.....	289
944	Figure D-23– TOU Information Received (No Billing) Snapshot Payload.....	289
945	Figure D-24– Block Information Delivered (No Billing) Snapshot Payload.....	290
946	Figure D-25– Block Information Received (No Billing) Snapshot Payload	291
947	Figure D-26– Format of the <i>GetSampledDataResponse</i> Command Payload	292
948	Figure D-27– Format of the <i>ConfigureMirror</i> Command Payload.....	293
949	Figure D-28– <i>MirrorReportAttributeResponse</i> Command Enabled	294
950	Figure D-29– <i>MirrorReportAttributeResponse</i> Command Disabled	295
951	Figure D-28– <i>ConfigureNotificationScheme</i> Command Payload	296
952	Figure D-29– <i>ConfigureNotificationFlags</i> Command Payload	297
953	Figure D-30– <i>Bit Field Allocation</i> Command Sub Payload.....	298
954	Figure D-31– <i>GetNotifiedMessage</i> Command Payload	299
955	Figure D-32– Supply Status Response Command Payload	300
956	Figure D-35– <i>StartSamplingResponse</i> Command Payload.....	301
957	Figure D-35– <i>NotificationScheme</i> Enumerations	306
958	Figure D-36– <i>Get Profile</i> Command Payload.....	307
959	Figure D-37– <i>Request Mirror Response</i> Command Payload	309
960	Figure D-38– <i>Mirror Removed</i> Command Payload	309
961	Figure D-39– <i>Request Fast Poll Mode</i> Command Payload	310
962	Figure D-40– <i>ScheduleSnapshot</i> Command Payload.....	311
963	Figure D-41– <i>SnapshotSchedulePayload</i> Format	311
964	Figure D-42– <i>TakeSnapshot</i> Command Payload	312
965	Figure D-43– <i>GetSnapshot</i> Command Payload	313
966	Figure D-44– <i>StartSampling</i> Command Payload	314
967	Figure D-45– <i>GetSampledData</i> Command Payload	315
968	Figure D-46– <i>MirrorReportAttributeResponse</i> Command Payload	316
969	Figure D-47– <i>ResetLoadLimitCounter</i> Command Payload	317
970	Figure D-48– Change Supply Command Payload	317
971	Figure D-49– Local Change Supply Command Payload	319
972	Figure D-50– <i>SetSupplyStatus</i> Command Payload	320
973	Figure D-51– <i>SetUncontrolledFlowThreshold</i> Command Payload	322
974	Figure D-52– Example of Data flow from IHD to Gas meter	326
975	Figure D-53– Price Cluster Client Server Example	332
976	Figure D-54– Single Threshold Set applied to All Consumption	349

977	Figure D-55– Single Threshold Set applied to Each Tier Consumption.....	349
978	Figure D-56– <i>GetCurrentPrice</i> Command Payload.....	361
979	Figure D-57– <i>Get Current Price</i> Command Options Field.....	361
980	Figure D-58– <i>GetScheduledPrices</i> Command Payload	362
981	Figure D-59– <i>PriceAcknowledgement</i> Command Payload	363
982	Figure D-60– <i>Get Block Period(s)</i> Command Payload.....	364
983	Figure D-61– <i>GetConversionFactor</i> Command Payload.....	365
984	Figure D-62– <i>GetCalorificValue</i> Command Payload	365
985	Figure D-63– <i>GetTariffInformation</i> Command Payload	366
986	Figure D-64– <i>GetPriceMatrix</i> Command Payload.....	367
987	Figure D-65– <i>GetBlockThresholds</i> Command Payload.....	368
988	Figure D-66– <i>GetCO₂Value</i> Command Payload	368
989	Figure D-67– <i>GetTierLabels</i> Command Payload.....	369
990	Figure D-68– <i>GetBillingPeriod</i> Command Payload	369
991	Figure D-69– <i>GetConsolidatedBill</i> Command Payload.....	370
992	Figure D-70– <i>CPPEventResponse</i> Command Payload.....	371
993	Figure D-71– <i>GetCreditPayment</i> Command Payload.....	372
994	Figure D-72– <i>Publish Price</i> Command Payload	375
995	Figure D-73– <i>Publish Block Period</i> Command Payload.....	385
996	Figure D-74– <i>PublishConversionFactor</i> Command Payload	388
997	Figure D-75– <i>PublishCalorificValue</i> Command Payload.....	389
998	Figure D-76– <i>PublishTariffInformation</i> Command Payload	390
999	Figure D-77– <i>PublishPriceMatrix</i> Command Payload	393
1000	Figure D-78– <i>PriceMatrix</i> Command Sub-Payload.....	394
1001	Figure D-79– <i>PublishBlockThresholds</i> Command Payload.....	395
1002	Figure D-80– <i>BlockThreshold</i> Sub-Payload Format	396
1003	Figure D-81– <i>PublishCO₂Value</i> Command Payload	397
1004	Figure D-82– <i>PublishTierLabels</i> Command Payload	399
1005	Figure D-83– <i>PublishBillingPeriod</i> Command Payload.....	400
1006	Figure D-84– <i>PublishConsolidatedBill</i> Command Payload	402
1007	Figure D-85– <i>PublishCPPEvent</i> Command Payload.....	403
1008	Figure D-86– CPP Event Flow.....	405
1009	Figure D-87– <i>PublishCreditPayment</i> Command Payload	406
1010	Figure D-88– <i>PublishCurrencyConversion</i> Command Payload	407
1011	Figure D-89– <i>CancelTariff</i> Command Payload	409
1012	Figure D-90– Messaging Cluster Client/Server Example.....	417
1013	Figure D-91– <i>Display Message</i> Command Payload	418
1014	Figure D-92– <i>Cancel Message</i> Command Payload	421
1015	Figure D-93– <i>Cancel All Messages</i> Command Payload.....	422
1016	Figure D-94– <i>Message Confirmation</i> Command Payload	423
1017	Figure D-95– <i>GetMessageCancellation</i> Command Payload	424
1018	Figure D-96– Client/Server Message Command Exchanges	425
1019	Figure D-97– A Client Requests a Tunnel From a Server to Exchange Complex	426

1020	Figure D-98– SE Device 1 (Client) Requests a Tunnel From SE Device 2 (Server) to Transfer Data Without Flow Control (Default)	428
1022	Figure D-99– SE Device 1 (Client) Requests a Tunnel From SE Device 2 (Server) to Transfer Data With Flow Control	429
1024	Figure D-100– <i>RequestTunnel</i> Command Payload	431
1025	Figure D-101– <i>CloseTunnel</i> Command Payload	433
1026	Figure D-102– <i>TransferData</i> Command Payload	433
1027	Figure D-103– <i>TransferDataError</i> Command Payload	434
1028	Figure D-104– <i>AckTransferData</i> Command Payload	435
1029	Figure D-105– <i>ReadyData</i> Command Payload	436
1030	Figure D-106– <i>Get Supported Tunnel Protocols</i> Command Payload	437
1031	Figure D-107– <i>RequestTunnelResponse</i> Command Payload	438
1032	Figure D-108– <i>TransferData</i> Command Payload	439
1033	Figure D-109– <i>Supported Tunnel Protocols Response</i> Command Payload	440
1034	Figure D-110– <i>Supported Tunnel Protocols Response</i> Command Protocol Fields	441
1035	Figure D-111– <i>TunnelClosureNotification</i> Command Payload	441
1036	Figure D-112– Prepay Cluster Client Server Example	444
1037	Figure D-113– <i>Select Available Emergency Credit</i>	471
1038	Figure D-114– Change Debt Command Payload	473
1039	Figure D-115– Emergency Credit Setup Command Payload	474
1040	Figure D-116– Consumer Top Up Command Payload	475
1041	Figure D-117– Credit Adjustment Command Payload	476
1042	Figure D-118– Change Payment Mode Command Payload	478
1043	Figure D-119– Get Prepay Snapshot Command Payload	479
1044	Figure D-120– Get Top Up Code Log Command Payload	480
1045	Figure D-121– Set Low Credit Warning Level Command Payload	480
1046	Figure D-122– GetDebtRepaymentLog Command Payload	481
1047	Figure D-123– Set Maximum Credit Level Command Payload	482
1048	Figure D-124– Set Overall Debt cap Command Payload	483
1049	Figure D-125– Publish Prepay Snapshot Command Payload	484
1050	Figure D-126– Debt/Credit Status Command Payload	486
1051	Figure D-127– Change Payment Mode Response Command Payload	487
1052	Figure D-128– Consumer Top Up Response Command Payload	489
1053	Figure D-130– Publish Top Up Log Command Payload	490
1054	Figure D-131– Top Up Code Payload	490
1055	Figure D-132– Publish Debt Log Command Payload	491
1056	Figure D-133– Format of the Debt Payload	491
1057	Figure D-134– Prepayment Credit Status Attribute Explained	493
1058	Figure D-135– Calendar Cluster	497
1059	Figure D-136– Recommended Calendar Command Sequence	499
1060	Figure D-137– <i>PublishCalendar</i> Command Payload	502
1061	Figure D-138– <i>PublishDayProfile</i> Command Payload	504
1062	Figure D-139– Schedule Entries for Rate Switch Times Command Sub-Payload	506

1063	Figure D-140– Schedule Entries for Friendly Credit Switch Times Command Sub-Payload....	506
1064	Figure D-141– Schedule Entries for Auxilliary Load Switch Times Command Sub-Payload...	507
1065	Figure D-142– <i>PublishWeekProfile</i> Command Payload.....	507
1066	Figure D-143– <i>PublishSeasons</i> Command Payload.....	509
1067	Figure D-144– Season Entry Sub-Payload.....	509
1068	Figure D-145– <i>PublishSpecialDays</i> Command Payload	510
1069	Figure D-146– <i>SpecialDayEntry</i> Sub-Payload	512
1070	Figure D-147– <i>CancelCalendar</i> Command Payload	512
1071	Figure D-148– <i>GetCalendar</i> Command Payload	513
1072	Figure D-149– <i>GetDayProfiles</i> Command Payload.....	514
1073	Figure D-150– <i>GetWeekProfiles</i> Command Payload	515
1074	Figure D-151– <i>GetSeasons</i> Command Payload	516
1075	Figure D-152– <i>GetSpecialDays</i> Command Payload.....	516
1076	Figure D-153– Device Management Cluster Client/Server Example.....	519
1077	Figure D-154– <i>RequestNewPassword</i> Command Payload	527
1078	Figure D-155– <i>Report Event Configuration</i> Command Payload	528
1079	Figure D-156– <i>Report Event Configuration</i> Sub-Payload	528
1080	Figure D-157– <i>Publish Change of Tenancy</i> Command Payload	529
1081	Figure D-158– <i>Publish Change of Supplier</i> Command Payload.....	531
1082	Figure D-159– <i>RequestNewPasswordResponse</i> Command Payload.....	532
1083	Figure D-160– <i>Update SiteID</i> Command Payload.....	533
1084	Figure D-161– <i>Set Event Configuration</i> Command Payload.....	534
1085	Figure D-162– ‘ <i>Apply by List</i> ’ Sub-Payload.....	535
1086	Figure D-163– ‘ <i>Apply by Event Group</i> ’ Sub-Payload.....	535
1087	Figure D-164– ‘ <i>Apply by Log Type</i> ’ Sub-Payload	536
1088	Figure D-165– ‘ <i>Apply by Configuration Match</i> ’ Sub-Payload	536
1089	Figure D-166– <i>Get Event Configuration</i> Command	536
1090	Figure D-160– <i>Update CIN</i> Command Payload	537
1091	Figure D-167– Event Cluster Client Server Example	566
1092	Figure D-168– Mirrored BOMD Event Cluster Client Server Example	567
1093	Figure D-168– <i>Get Event Log</i> Command Payload	568
1094	Figure D-169– <i>Clear Event Log Request</i> Command Payload	570
1095	Figure D-170– <i>Publish Event</i> Command Payload	571
1096	Figure D-171– <i>Publish Event Log</i> Command Payload	572
1097	Figure D-172– <i>Publish Event Log</i> Sub-Payload	573
1098	Figure D-174– <i>Clear Event Log Response</i> Command Payload	574
1099	Figure D-173– Energy Management Cluster	576
1100	Figure D-174– <i>Manage Event</i> Command Payload	581
1101	Figure D-175– <i>Report Event Status</i> Command Payload	583
1102	Figure D-176– MDU Pairing Cluster.....	585
1103	Figure D-177– <i>Pairing Response</i> Command Payload.....	586
1104	Figure D-178– <i>Pairing Request</i> Command Payload	588
1105	Figure E-1– Smart Energy Device Class Reference Example.....	592

1106	Figure E-2– Correctly Overlapping Events	592
1107	Figure E-3– Correct Superseding of Events	593
1108	Figure E-4– Superseded Event for a Subset of Device Classes	594
1109	Figure E-5– Ending Randomization Between Events.....	595
1110	Figure E-6– Start Randomization Between Events	596
1111	Figure E-7– Acceptable Gaps with Start and Stop Randomization.....	597
1112		

1113 Participants

1114

1115 Contact Information

1116

1117 Much of the information in this document is preliminary and subject to change.
1118 Members of the ZigBee Working Group are encouraged to review and provide inputs for this
1119 proposal. For document status updates, please contact:

1120

1121 Ian Winterburn

1122 Landis+Gyr

1123 Orion Business Park, Bird Hall Lane,
1124 Stockport, Cheshire, SK3 0RT, UK

1125

1126 E-Mail: ian.winterburn@landisgyr.com

1127 Phone: +44 161 367 5356

1128

1129 You can also submit comments using the ZigBee Alliance reflector. Its web site address is:

1130 www.zigbee.org

1131

1132 The information on this page should be removed when this document is accepted by the Working
1133 Group.

1134

1135

1136 **Participants**

1137
1138 The following is a list of those who were members of the ZigBee Architecture Review
1139 Committee (ZARC) leadership when this document was released:

1140 **Skip Ashton:** Chair

1141 **Phil Jamieson:** Vice-Chair

1142
1143 When the document was released, the Smart Energy Application Profile Work Group
1144 leadership was composed of the following members:

1145 **Larry Kohrmann:** Chair

1146 **Ian Winterburn:** Vice-Chair

1147 **Rob Alexander & Ian Winterburn:** Technical Editors

1148 **Jeff Shudark:** Secretary

1149 Contributions were made to this document from the following members:

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

1150

Document History

1151

1152 Table 1.1 shows the change history for this specification.

1153

1154

Table 1-1 – Document Revision Change History

Revision	Version	Description
--	0.9	Original version (based on 14-0163-04).
0	0.9	<p>Initial changes resulting from comment resolution for test events #4 to #10, including:-</p> <p>Updates to handling of new cryptographic curve</p> <p><i>GetNotificationFlag</i> command renamed <i>GetNotifiedMessage</i></p> <p>Prepayment cluster <i>Get Command</i> command removed (Metering cluster <i>GetNotifiedMessage</i> command to be used instead)</p> <p>Clarification of mirror configuration commands</p> <p>Mirror Configuration attribute set removed from Metering cluster</p> <p><i>SetSupplyStatus</i> command fields changed from Booleans to Enumerations</p> <p><i>SampleRequestInterval</i> field added to <i>GetSampledDataResponse</i> command</p> <p>Events cluster section updated (Annex D.11)</p>
1	0.9	<p>Metering cluster Sampling functionality updated in line with the requirements of comment #TE11-1.</p> <p><i>Calendar Time Reference</i> re-introduced in the <i>PublishCalendar</i> command for comment #WG-1.</p> <p>Changes for comment #TE9-8 extended to cover <i>ChangeofSupplier</i> as well as <i>ChangeofTenancy</i> (omission from original changes).</p> <p>Modifications for comments #TE4-6, #TE6-4, #TE7-27, #TE7-43, #TE7-48, #TE9-1, #TE10-1, #TE10-9, #TE10-10, #TE10-11, #TE10-12, #TE10-13, #TE10-30 and #WG-2.</p>
2	0.9	PDF version of r01.

1155

1156

1157

1 Introduction

1158

1.1 Scope

1159

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

1160

1161

1162

1163

1164

1165

1166

1.2 Purpose

1167

1168

1169

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

1170

1.3 Provisional Features

1171

1172

1173

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

1174

- Metering cluster Historical Consumption attributes 0x09-0x0E, 0x11 and 0x12

1175

- Metering cluster Block Information attribute set.

1176

- Metering cluster Alarms attribute set.

1177

- Price cluster attribute sets 0x01-0x04, which include Block Threshold, Block Period, Commodity, and Block Price Information attribute sets.

1178

- Price cluster Price Tier values above 0x06.

1179

- Price cluster *Get Block Price* command.

1180

- Price cluster *Publish Block Period* command.

1181

- The Block Threshold field of the *Publish Price* command.

1182

- The Price cluster client attributes.

1183

- Block Pricing application guidelines.

1184

- Tunneling cluster Flow Control option.

1185

- Prepayment cluster.

1186

- Trust Center Swapout behaviors.

1187

1188

1189

2 References

1190

2.1 References

1191

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

1197

1198

2.1.1 ZigBee Alliance Documents

1199

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

1200

[B2] ZigBee document 064309r04, Commissioning Framework

1201

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

1202

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

1203

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

1204

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

1205

[B7] ZigBee document 08006r03, ZigBee 2007 Layer PICS and Stack Profiles,
ZigBee Core Stack Working Group

1206

[B8] Over the Air Upgrade Cluster Spec. 09-5264-19

1207

[B9] Over the Air Upgrade Cluster test spec: 09-5473-06

1208

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

1209

2.1.2 External Reference Documents

1210

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

- 1232 [B12] ANSI X9.62-2005, Public Key Cryptography for the Financial Services Industry:
1233 The Elliptic Curve Digital Signature Algorithm (ECDSA), American Bankers Association.
1234 Available from <http://www.ansi.org>.
- 1235
- 1236 [B13] ANSI X9.63-2001, Public Key Cryptography for the Financial Services Industry -
1237 Key Agreement and Key Transport Using Elliptic Curve Cryptography, American Bankers
1238 Association, November 20, 2001.
1239 Available from <http://www.ansi.org>.
- 1240
- 1241 [B14] NIST Special Publication 800-56A, Recommendation for Pair-Wise Key
1242 Establishment Schemes Using Discrete Logarithm Cryptography (Revised), March 2007.
1243 Available from <http://csrc.nist.gov>.
- 1244
- 1245 [B15] NIST Special Publication 800-38C, Recommendation for Block Cipher Modes of
1246 Operation: The CCM Mode for Authentication and Confidentiality, May 2004.
1247 Available from <http://csrc.nist.gov>.
- 1248
- 1249 [B16] FIPS Pub 197, Advanced Encryption Standard (AES), Federal Information
1250 Processing Standards Publication 197, US Department of Commerce/N.I.S.T., Springfield,
1251 Virginia, November 26, 2001.
1252 Available from <http://csrc.nist.gov>.
- 1253
- 1254 [B17] FIPS Pub 198, The Keyed-Hash Message Authentication Code (HMAC), Federal
1255 Information Processing Standards Publication 198, US Department of Commerce/N.I.S.T.,
1256 Springfield, Virginia, March 6, 2002.
1257 Available from <http://csrc.nist.gov>.
- 1258
- 1259 [B18] Standards for Efficient Cryptography: SEC 1 (working draft) ver 1.7: Elliptic Curve
1260 Cryptography, Certicom Research, November 13, 2006.
1261 Available from <http://www.secg.org>
- 1262
- 1263 [B19] Standards for Efficient Cryptography: SEC 4 (draft) ver 1.0: Elliptic Curve
1264 Cryptography, Certicom Research, January 24, 2013.
1265 Available from <http://www.secg.org>
- 1266
- 1267 [B20] RFC 3280: Internet X.509 Public Key Infrastructure: Certificate and Certificate
1268 Revocation List (CRL) Profile. IETF, April 2002.
1269 Available from <http://www.ietf.org>
- 1270
- 1271 [B21] DUPLICATE OF [B19] - REMOVED
- 1272
- 1273 [B22] RFC 3280: Internet X.509 Public Key Infrastructure: Certificate and Certificate
1274 Revocation List (CRL) Profile. IETF, April 2002. Available from <http://www.ietf.org>
- 1275
- 1276 [B23] Standards for Efficient Cryptography: SEC 2 (Final) ver 2.0: Recommended Elliptic
1277 Curve Domain Parameters

1278

3 Definitions

1279

3.1 Conformance Levels

1280
1281
1282
1283

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.

1284
1285
1286

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

1287
1288
1289

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”).

1290
1291
1292
1293

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”).

1294

1295

3.2 ZigBee Definitions

1296
1297
1298

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

1299
1300

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

1301
1302
1303
1304

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.

1305
1306
1307
1308
1309
1310

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.

1311
1312

Node: Same as a unit.

1313
1314
1315

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.

1316
1317

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

1318
1319
1320

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.

1321
1322 **ZigBee coordinator:** An IEEE 802.15.4-2003 PAN coordinator.
1323
1324 **ZigBee end device:** an IEEE 802.15.4-2003 RFD or FFD participating in a ZigBee network,
1325 which is neither the ZigBee coordinator nor a ZigBee router.
1326
1327 **ZigBee router:** an IEEE 802.15.4-2003 FFD participating in a ZigBee network, which is not
1328 the ZigBee coordinator but may act as an IEEE 802.15.4-2003 coordinator within its
1329 personal operating space, that is capable of routing messages between devices and supporting
1330 associations.

1331 **3.3 Smart Energy Definitions**

1332 **Delivered:** In the context of metering, the term “Delivered” refers to the quantity of a commodity
1333 that was delivered to the customer from the utility.

1334 **Export:** In the context of metering, the term “Export” has the same meaning as “Received”.

1335 **Friendly Credit Period:** A duration of time where delivery rules for supply interruption
1336 connected with the accounting functions of a meter are temporarily suspended. Also known as
1337 “Non-Disablement Period”. **NOTE1** - When a meter enters into a Friendly Credit Period with a
1338 useable positive credit balance, the consumer will be allowed to consume energy for the duration
1339 of the Friendly Credit Period, regardless of their credit status while in that period. If, however,
1340 the consumer had already run out of credit and supply was interrupted before entering into the
1341 Friendly Credit Period, they shall not be allowed to reconnect without first adding suitable
1342 additional credit. **NOTE2** - At the end of the Friendly Credit Period, the normal delivery rules
1343 connected with the accounting functions of the meter shall be resumed, and if the meter’s credit
1344 balance has dropped below the disablement threshold during the Friendly Credit Period, then the
1345 meter shall disconnect upon resuming normal delivery rules.

1346 **Import:** In the context of metering, the term “Import” has the same meaning as “Delivered”.

1347 **Received:** In the context of metering, the term “Received” refers to the quantity of a commodity
1348 that was received by the utility from the customer.

1349 **Top-up:** An operation or transaction resulting in the credit balance held on a prepayment meter
1350 to be increased by use of a credit token. **NOTE** - In IEC terminology, the associated term ‘vend’
1351 would normally relate to a transaction in conjunction with a vending system at a point of sale,
1352 resulting in the creation of a token that can be transported by means of a physical or virtual token
1353 carrier.

1354 **Unique Transaction Reference Number (UTRN):** A numeric or alpha-numeric code used as a
1355 virtual token carrier for a payment metering system. These codes are issued by points of sale and
1356 represent an encoded and secured credit value or special action that the meter must act upon
1357 within its own application code. The functionality of payment metering tokens is outside the

1358 scope of the Smart Energy standard, however further information and examples of such tokens
1359 can be seen in IEC 62055-41.

1360

1361

1362

4 Acronyms And Abbreviations

1363

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

1364

1365

1366

1367

5 Profile Description

1368

1369

5.1 A ZigBee Smart Energy Network

1370

1371 The Smart Energy market requires two types of ZigBee networks for metering and energy
1372 management. These include neighborhood area networks for meters, using ZigBee for sub-
1373 metering within a home or apartment, and using ZigBee to communicate to devices within
1374 the home. Different installations and utility preferences will result in different network
1375 topologies and operation and this profile must allow for these differences. However, each of
1376 these networks will operate using the same Basic Principles to ensure interoperability.
1377

1378

1379 Because of the type of data and control within the Smart Energy network, application
1380 security is a key requirement. The application will use link keys which are optional in the ZigBee
1381 and ZigBee Pro stack profiles but are required within a Smart Energy network. The Trust Center
1382 and all devices on the Smart Energy network must support the installation and use of these keys
1383 as described in the security section.

1384

1385 Metering networks are primarily installed by specialized service personnel, but other devices in
1386 the network may be added by home owners, or home automation professionals who may not
1387 have any ZigBee expertise. Installation concepts must be easy and uniform across Smart Energy
1388 device manufacturers.

1389

1390 Smart Energy networks could include both ZigBee 2007 and ZigBee 2007 Pro nodes. It is
1391 recommended the majority of the nodes in the network should be based on one stack profile or
1392 the other to get consistent performance. ZigBee Smart Energy certified products must be
1393 based upon a ZigBee Compliant Platform (ZCP). If the Smart Energy profile resides in
1394 conjunction with a private profile, the product should be ZigBee Manufacturer Specific
1395 Profile (MSP) certified and must be Smart Energy ZCP certified. This additional certification
1396 provides a reassurance that the underlying stack is behaving properly and the application is not
1397 abusive to the network.

1398

1399 Smart Energy networks will not interact with a consumer ZigBee Home Area Network unless a
1400 device is used to perform an “application level bridge” between the two profiles or the HA
1401 devices satisfy the Smart Energy profile security requirements. This is due to the higher security
1402 requirements on the Smart Energy network that are not required on a Home network. However,
1403 it is expected that Home Automation devices that are extended to include the Smart Energy
1404 profile can still operate in a home network.

1405

1406 The ZigBee Smart Energy Network makes possible networks such as the following:

1407
1408
1409
1410
1411
1412
1413



Utility Private HAN

- Customer
- Utility
- Shared

Energy Service Interface



Load Control Device

1422

Figure 5-1 Utility Private HAN

1423

1424

1425 Utility Private HAN might include an in-home display, or a load control device working in
1426 conjunction with an energy service interface, but it would not include any customer controlled
1427 devices.

1428

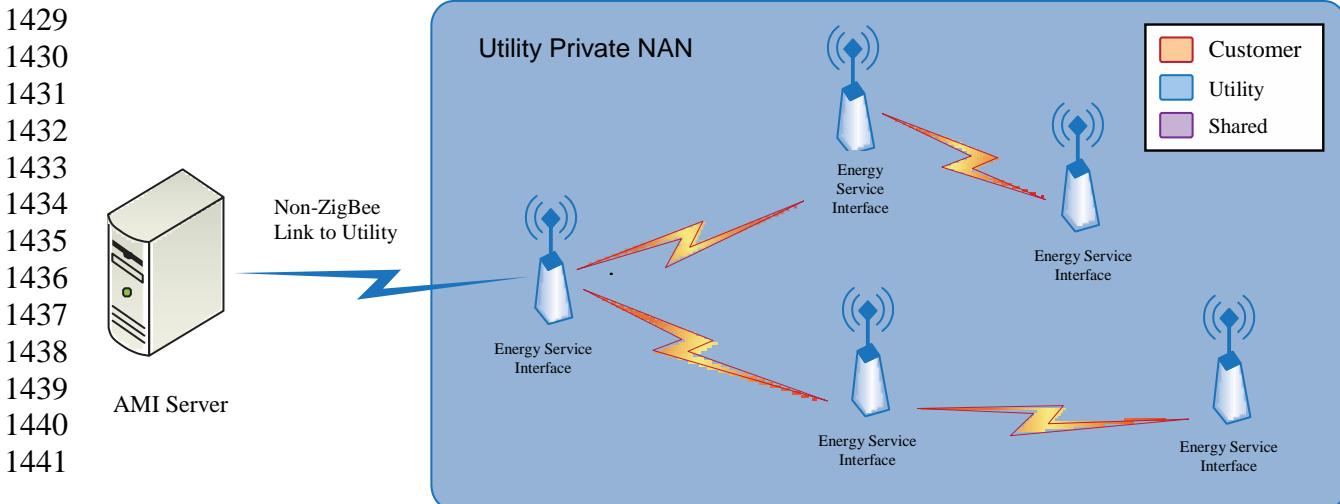


Figure 5-2 Utility Private NAN

1442
1443
1444 Utility Private ZigBee network might also be used as a NAN, where ZigBee provided the
1445 primary communications for a Smart Energy deployment.
1446
1447
1448
1449
1450
1451

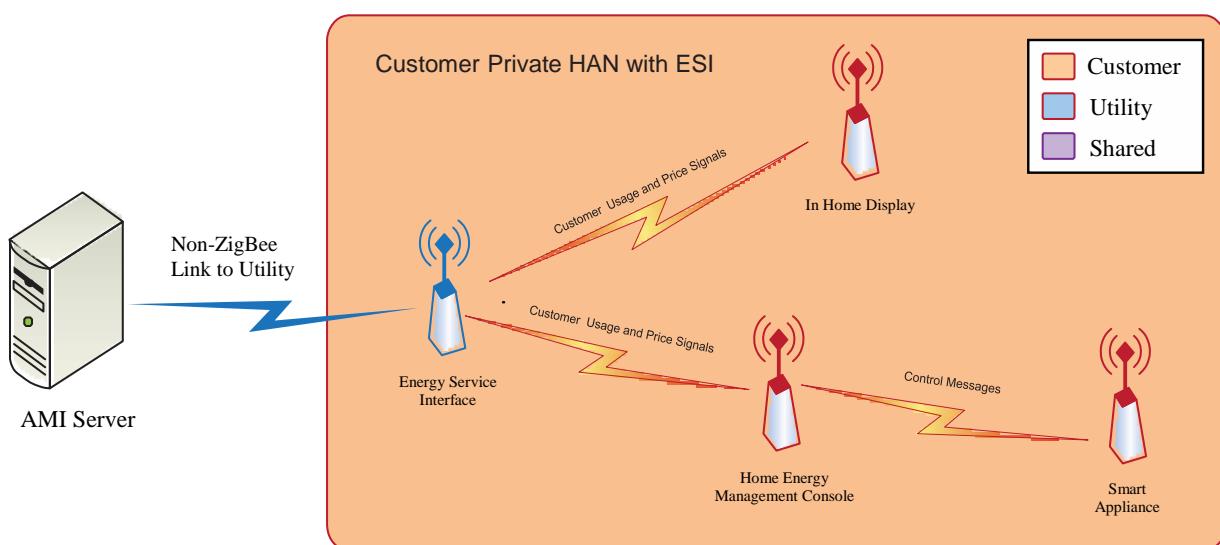


Figure 5-3 Customer Private NAN

1461
1462
1463 ESI provided by utility, but limited to the role of information provider (Usage and Pricing)
1464 into a customer HAN that utilizes an Energy Management Console for conveying or
1465 controlling local devices. An example is controlling a smart appliance based upon a pricing
1466 signal.
1467

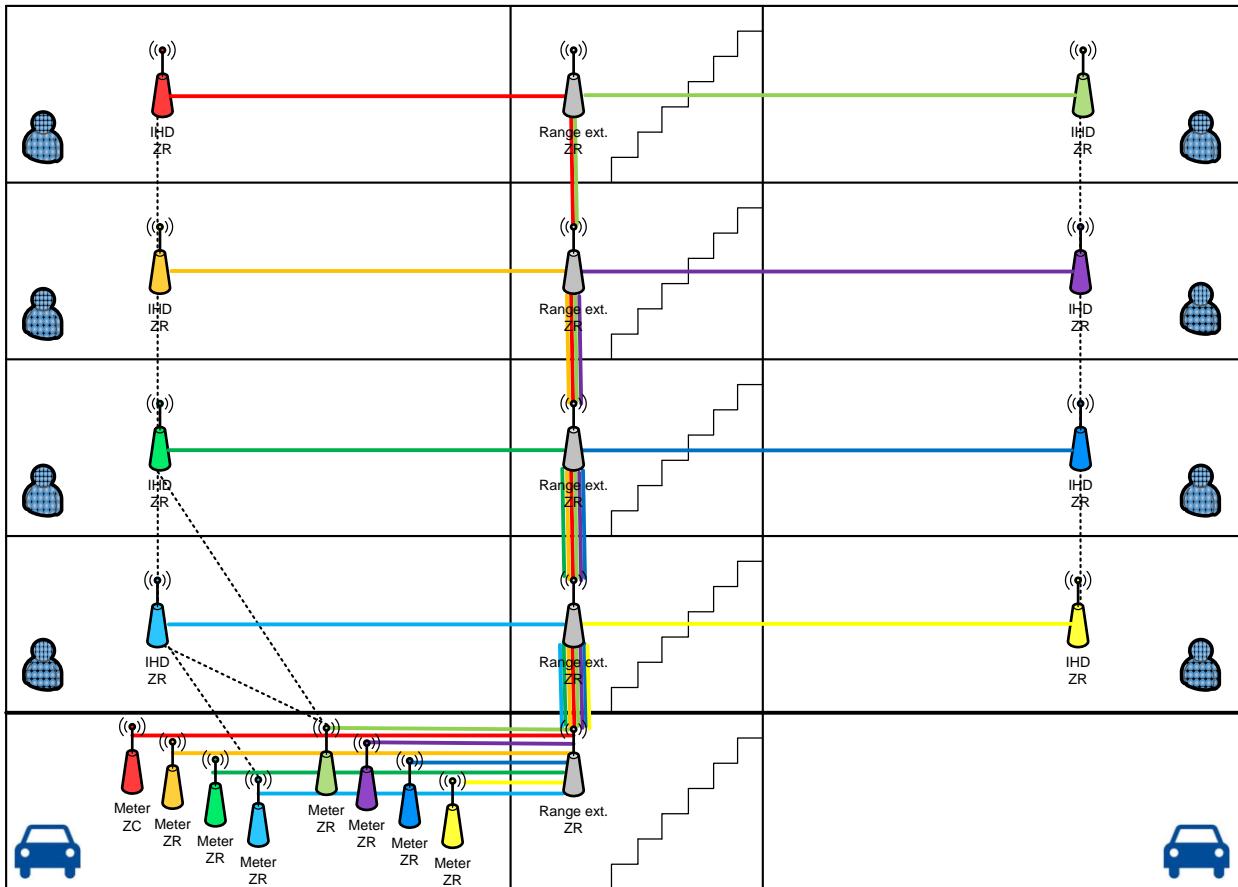


Figure 5-4 An Example of a MDU Setup

1468
1469

1470

1471 Figure 5-4 shows an example of a MDU setup, with multiple apartments in the same building
 1472 (each hosting a number of ZigBee Smart Energy devices including IHDs, thermostats, range
 1473 extenders etc.) and a so-called *meter room*, normally located in the basement, where electricity
 1474 meter/ESI devices are installed. Range extenders are shown as well in the main stairway of the
 1475 building to reinforce the mesh. The ‘Federated’ Trust Center (shown as Meter ZC in the diagram)
 1476 provides the single back-haul connection in this example.
 1477

1478

5.2 ZigBee Stack Profile

1479 Products that conform to this specification shall use stack profile number 0x01 or profile 0x02, as
 1480 defined in [B7]. In addition to the requirements specified in [B7], the following requirements are
 1481 mandatory for this application profile.

1482

- 1483 • Support for Application link keys is required.

- 1484 • Fragmentation is required. Please refer to 5.3.8 regarding fragmentation sizes and parameter
1485 settings.

1486

1487 **5.2.1 MAC Data Polling (NLME_Requests)**

1488 MAC Data polling is required by all sleepy end devices to operate correctly in a ZigBee Pro
1489 network. Smart Energy puts no restrictions on the frequency of MAC data polls. The choice of
1490 how frequently data polling is done will be based on individual product design considerations
1491 to reduce power consumption. However the following are a set of recommendations to ensure
1492 correct operation in the network:

1493

- 1494 • The MAC data polling rate should be dynamic based on the device's operating state. It is
1495 recommended it has at least two rates, a fast rate and a slow rate.

1496

- 1497 • The ZigBee specification only requires that parent devices buffer a single message for 7.5
1498 seconds. This single buffer applies to all sleepy end devices. Therefore a sleepy device
1499 should poll more frequently than once per 7.5 seconds in order to be able to retrieve a
buffered message.

1500

- 1501 • When the device is waiting for an active response message such as an APS
1502 acknowledgement, or a ZCL response, or participating in a multi-message protocol, it should
poll at its fast rate. This fast rate is recommended to be at least once every 3 seconds.

1503

- 1504 • When the device is not actively waiting for messages it can poll at its slow rate. For example
1505 once per hour. This ensures it still has a connection with the network and with its parent.

1506

1507 During initial joining to the Smart Energy network, including key establishment and service
discovery, it should poll at its fast rate.

1508

1509 **5.2.2 Application Level Queries**

1510

1511 It is expected that client devices will periodically send application level queries to servers to
1512 retrieve data. This may be done for example by thermostats querying the current price, or an in-
1513 home display to show the current reading of a meter.

1514

1515 Due to the fact that all ZigBee devices within the HAN utilize a shared medium for sending and
1516 receiving data it is recommended that devices do not saturate the network with frequent queries
1517 for data that does not change often. As a general rule, but not a requirement, it is
1518 recommended that devices do not initiate more than 1 query per second. This
1519 recommendation does not apply to responses generated locally due to the receipt of remote
1520 device requests. In addition, it is possible that the device may need to generate a burst of

1521 traffic and exceed this recommendation. This bursting period should be very limited and
1522 followed by a period of reduced traffic respecting the above guidelines.
1523

1524 **5.2.3 ZigBee Coordinator and Trust Center Recommendations**

- 1525
- 1526 • In a Smart Energy based HAN network the Trust Center shall be the Coordinator (short
1527 address 0x0000).
 - 1528 • In a Smart Energy based HAN network the Trust Center shall be an ESI in the network.
 - 1529 • In a Smart Energy based NAN the backhaul point is likely to be the coordinator and trust
1530 center.
- 1531

1532 **5.3 Startup Attribute Set (SAS)**

1533 In order to ensure interoperability, all ZigBee Smart Energy devices shall implement
1534 compatible Startup Attribute Sets (SAS) as defined in this specification. This does not
1535 mean that the set must be modifiable through a commissioning cluster, but that the device
1536 must internally implement these stack settings to ensure compatibility and consistent user
1537 experience. The startup set parameters described by the commissioning cluster in [B2] provide
1538 a good basis to specify a Smart Energy start up set.
1539

1540 Because Smart Energy Devices are likely to be preconfigured at a warehouse and installed by a
1541 technician, specific start up set values may be established by a particular utility or service area
1542 and these startup set values used in place of those below for installation. The startup set values
1543 that would be expected to be set by the installer are noted below.
1544

1545 **5.3.1 Startup Parameters**

1546 The startup parameters and their default values are listed in Table 5-1.
1547

1548 **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
-------------------	--------------	--

1549

5.3.2 Join Parameters

1551 The join parameters and their default values are listed in Table 5-2.

1552

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.
--	-------------------	------------	--

1553

5.3.3 Security Parameters

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

1556

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.

1557

5.3.4 End Device Parameters

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

1560

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.

1561

5.3.5 Link Status Parameters

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

1564

Table 5-5 – Link Status Parameters

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

1565

1566

1567

5.3.6 Concentrator Parameters

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

1570

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.

1571

5.3.7 APS Transport Parameters

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

1574

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.

1575

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.

1579

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).

1580

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.

1590

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

1593

5.3.9 Binding Parameters

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

1596

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.

1597

1598 **5.4 Smart Energy Profile Security**

1599 To be part of a Smart Energy network, a device shall associate using one of the two
 1600 association methods described below and require the use of the Key Establishment
 1601 Cluster (see Annex C) for installation and updating of link keys.
 1602

1603 All devices shall have the ability to retain their joining and security settings through power
 1604 outages.
 1605

1606 **5.4.1 Joining with Preinstalled Trust Center Link Keys**

1607 When using preinstalled trust center link keys, the following steps are used:

- 1608 1 Trust Center link keys SHALL be installed in each device prior to joining the utility network.
- 1609 2 The trust center link key for a device that is to be joined SHALL be provided to the local trust
 1610 center through an out of band means as described in sub-clause 5.4.8.1 “Out of Band Pre-
 1611 Configured Link Key Process”.
- 1612 3 Permit joining is turned on in the network. The Trust Center enables joining by calling the
 1613 NLMEPERMIT-JOINING.request primitive. Joining must be managed for an appropriate
 1614 amount of time but SHALL NOT be broadcast with a time of greater than 254 seconds should
 1615 not repeatedly broadcast without hearing device announcement or network administrator
 1616 action. The appropriate amount of time will be dictated by the overall performance of the
 1617 system and business processes driving the registration and device authorization activities. See
 1618 sub-clause 5.4.1.2, “Best Practice for Coordinator Permit Joining Broadcasts”.
- 1619 4 Be aware Joining has an internal time out within the ZigBee stack, therefore joining may need
 1620 to be enabled multiple times during the overall Registration and device authorization process.
- 1621 5 A device autonomously joining a network (i.e. without user supervision or input) may initially
 1622 scan for networks to join three times in succession without pausing. After failing to
 1623 successfully join a network, the device SHALL exponentially increase time between scan
 1624 times, eventually performing a channel scan at a maximum rate of once per hour. The device
 1625 may increase scan rate upon request from user input, such as a button push or power cycle.

- 1626 **6** The device joins the network and is sent the network key encrypted with the key-transport key
1627 derived for the preinstalled trust center link key. The procedure for doing this is detailed in
1628 Annex F, also reference [B3] section 4.5.4 on key-transport keys and [B3] section 4.4.1 on
1629 frame security for the APS layer.
- 1630 **7** After completion of the joining procedure, the device must use the Key Establishment Cluster
1631 to establish a new link key with the trust center. The device shall use its security credentials
1632 when initiating the key establishment protocols.
- 1633 **8** The trust center of the network has the option of later updating the trust center link keys with
1634 devices in the network as desired by the application using the Key Establishment Cluster.
1635 Updating security keys should be an infrequent operation.
- 1636 **9** Once joining is completed, the list of authorized devices in the Trust Center should be updated,
1637 please refer to sub-clause 5.4.1.1, “Best Practices for Tracking Registered Devices”.

1638
1639
1640 **5.4.1.1 Best Practices for Tracking Registered Devices**

1641 In order to properly track Smart Energy Devices and communicate device registration
1642 status to upstream systems, Trust Centers (ESIs) should maintain a list of authorized devices. It
1643 is also recommended that Trust Centers maintain the following items for each of the registered
1644 devices:

1645
1646
1647 **1** Client EUI64

1648 **2** Client Installation Code

1649 **3** Registration Status

1650 **4** Time and Date Stamps

1651 **5** Supported Cryptographic Suites

1652 With the exception of the cryptographic suite, this information is not exposed through the
1653 ZigBee network. However, the device information is expected to be used to track and
1654 understand ZigBee network connectivity.

1655
1656 **5.4.1.2 Best Practice for Coordinator Permit Joining Broadcasts**

1657 It will be left to the coordinator / administrators of the network to determine when a network
1658 should be allowing joining. However when the network is allowing joining:
1659

- 1660 1 At the start of the joining period the coordinator will allow joining and broadcast a permit join
1661 message for the lesser of the permit join period or 254 seconds.
- 1662 2 Every 240 seconds or whenever a device announce is received the coordinator will broadcast a
1663 permit join message for the lesser of the remaining permit join period or 254 seconds.
1664 Administrators of a network shall try to keep the amount of time devices on their networks
1665 allow joining to a minimum.

1666 *Note: sending out a permit join message with a time of 255 (forever) is disallowed due to the risk
1667 of not being able to reliably tell devices to stop permitting joining in the future.*
1668

1669 5.4.2 Re-Joining a Secured Network

1670 5.4.2.1 Rejoining Node Operation

1672 When a device is re-joining a secured network, the following steps are used:

1673 1 Permit joining is not required to be on in the network.

1675 2 The device shall attempt a rejoin using the procedure detailed in [B3] Section 3.6.1.4.2 with
1676 network security. The network key and sequence number used will be the ones previously
1677 obtained from the trust center.

1678 3 If the secured rejoin is successful, nothing more is required from the device.

1679 4 If the secured rejoin fails, the device shall attempt a rejoin using the procedure detailed in [B3]
1680 Section 3.6.1.4.2 without network security. The re-joining device is assumed to have
1681 previously joined the network and obtained a link key using the key establishment cluster
1682 procedures. If the device does not have a link key obtained via the key establishment cluster, it
1683 cannot rejoin the network.

1684 5 If the rejoin fails the device may attempt it again. If the device is told to leave the network it
1685 may employ the Joining using the Key Establishment Cluster procedure.
1686

1687 5.4.2.2 Trust Center Operation

1688 When the trust center receives notification that a device has rejoined the network, the following
1689 steps are used:

1690 1 If the device performed a secured rejoin the trust center is not required to take any action.
1691

1692 **2** If the device performed a rejoin the trust center shall determine if the device is authorized to be
1693 on the network. The trust center should send out an updated copy of the network key encrypted
1694 with the corresponding link key.

1695 **3** If the trust center determines that the device is not authorized to be on the network, it shall
1696 send an APS *Remove Device* command to the parent of the rejoicing device, with the target
1697 address of the rejoicing device's IEEE address. The parent will then remove that device from
1698 its child table.

1699
1700 *Note: The Trust Center and Router behaviors described in sections beginning at sub-clause
1701 5.4.2.2.1 up until sub-clause 5.4.3 in this revision of this specification are provisional and not
1702 certifiable. This text may change before reaching certifiable status in a future revision of this
1703 specification.*

1704

1705 **5.4.2.2.1 Initiating Re-Registration**

1706 To initiate the re-registration process for a device, the Trust Center (ESI) would invalidate the
1707 Link keys for that device and subsequently cause a re-authentication / authorization to
1708 re-establish Link Keys. The processes required for this activity are:

1709

1710 **1** The Trust Center invalidates the Link key by using the APSME-SET primitive.

1711 **2** When the Client device detects communication errors due via APS error results or by
1712 experiencing multiple re-try failures, both caused by the invalid Link Keys, it starts the
1713 processes to validate the following conditions:

1714 **a** The Device validates its still part of the network.

1715 **b** Route discovery processes validate communications paths are still in place.

1716 **3** If both conditions are true, the Client device attempts a secure re-join outlined in Re-joining a
1717 Secured Network and subsequently refreshes the Link Keys.

1718 **4** Re-binding of services take place (if needed).

1719 **5** Once Registration is completed, the list of authorized devices in the Trust Center should be
1720 updated, please refer to sub-clause 5.4.1.1.

1721

1722 **5.4.2.2.2 Initiating De-Registration**

1723 To initiate the de-registration process for a device, which is the process of removing a
1724 previously registered device, the Trust Center (ESI) would use the following processes for this
1725 activity:

- 1726 **1** The Trust Center (ESI) invalidates the Link key by using the APSME-SET primitive.
- 1727 **2** The Trust Center (ESI) informs the Client device to leave the network by calling the NLME-
1728 LEAVE.request primitive.
- 1729 **3** The Trust Center (ESI) informs any Routers to remove the Client device by calling the
1730 APSME-REMOVEDevice.request
- 1731 **4** The ESI would unbind any services associated with the Client device by calling the APSME-
1732 UNBIND primitive.
- 1733 **5** Once de-registration is completed, the list of authorized devices in the Trust Center should be
1734 updated.

1735

1736 **5.4.2.2.3 Trust Center Swap-Out**

1737 *Note: The Trust Center Swap-Out feature in this revision of this specification is provisional
1738 and is not certifiable. This feature set may change before reaching certifiable status in a future
1739 revision of this specification.*

1740 This section describes the requirements for swapping out a Trust Center in a Smart Energy
1741 network. In the Smart Energy Standard, an ESI should act as the coordinator and trust center of
1742 the network. In most deployments the ESI is the meter and therefore the TC. There can only
1743 be one TC in a SE network, although multiple ESIs may exist on the network. The TC (ESI) in
1744 a SE network is responsible for performing authentication and authorization. SE devices which
1745 are allowed to join the network are provisioned on the TC (ESI) from the head-end over the
1746 utility's backhaul connection.

1747 When a TC is replaced the new device is given the extended PAN ID of the previous
1748 network, and the addresses and associated trust center link keys of all the devices from the
1749 previous network. Both the existing devices and the TC treat these keys like installation codes
1750 (unauthorized), which have limited privileges in the network. Once the devices successfully
1751 connect to the new trust center they must re-establish new TC link keys using CBKE.

1752 Existing Smart Energy devices must be upgraded to include behavior that allows them to
1753 detect a failure to communicate with the existing trust center. When it detects this condition a
1754 device will go off in search of another network with the same extended PAN ID as the current
1755 one. If a network is found then the device will perform a first time join using the NWK rejoin
1756 and its current TC link key as the pre-configured key. If the device is able to successfully join
1757 the network then it will immediately initiate CBKE to derive a new link key with the
1758 replacement TC. If that succeeds the device will identify the device's IEEE as the identity of the
1759 new trust center, and begin operating in the new network; the device shall locate any services
1760 that it may have been using.

1761 If it is unable to join to the new network or unable to successfully negotiate CBKE, then the
1762 device will return to its previous network and continue operating. If the trust center is still
1763 unreachable at a later point in time it can perform the above steps again to attempt to find a
1764 new network.

1765 Trust Center Swap-Out is an optional feature and is not required for ZigBee Smart Energy
1766 Certification.

1767

1768 **5.4.2.2.3.1 SE Router Requirements**

1769 All routers in the network shall be able to identify when the trust center is no longer
1770 accessible in the network. This will be done by periodically sending an APS datagram to the
1771 Trust Center and receiving the APS acknowledgment. The APS datagram shall require
1772 encryption and acknowledgement.

1773 After an extended period where multiple attempts have been made to contact the trust center and
1774 failed to get a response, a device would temporarily drop off the network to go in search of a
1775 network where the trust center was present. The new network may have different network
1776 parameters than the old one, but the extended PAN ID value would always be the same. Those
1777 networks that match all of the parameters of the old network will be filtered out in preference of
1778 a new network with one or more different parameters. This enables the device to find a newer
1779 instance of the existing network.

1780 Once a new instance of the existing network has been found, the device would perform
1781 various procedures to attempt to join that network and authenticate with the new trust center
1782 device. If at any point during the attempt a failure occurred, the device may continue scanning
1783 for networks to join or return to its existing network and continue operating as it had before.

1784 As a last resort, all devices must have a means to return to factory defaults so that they can be
1785 recommissioned. This would involve reverting back to the use of an installation code and
1786 forgetting all previous network and application parameters. This provides a means to reconnect
1787 the device to the existing network when other methods have failed, or decommission the device
1788 and join it to a new network.

1789

1790 **5.4.2.2.3.2 Per SE Network Storage Requirements**

1791 TC swap-out requires the backup of data to an off-chip device. The data and storage
1792 requirements are listed in Table 5.10. Backup of the Extended PAN ID should be performed
1793 once the ESI has been commissioned or the network is formed. Backup of the TC Link Key
1794 Hash (see sub-clause 5.4.2.2.3.6) should be performed on successful completion of CBKE with
1795 the TC. TC Link Key updates from subsequent CBKE shall also be backed up. The Install
1796 Code derived TC Link Key may be backed up when the device is provisioned on to the SE
1797 network.

1798

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

1799

1800

1801

1802 **5.4.2.2.3.3 Utility Requirements**

1803 It is expected that the utility is able to store backup data about each Smart Energy network in
 1804 order to facilitate the TC swap-out feature. It is recommended that the list of IEEE (EUI64)
 1805 addresses of devices registered in the network, and their associated installation code, always be
 1806 backed up. This will help to deal with an unexpected situation due to the customer or the utility,
 1807 which requires one or more HAN devices to be recommissioned.

1808

1809 **5.4.2.2.3.4 Keep Alive Method**

1810 In order to detect the TC is no longer available all SE routers shall implement a keep-alive
 1811 mechanism with the TC. The Key Establishment cluster is mandatory on all SE devices. The SE
 1812 routers shall send an APS encrypted ZCL message on a periodic interval of up to a maximum of
 1813 20 minutes. The minimum polling rate should not be less than 5 minutes. Failure to receive an
 1814 encrypted APS data frame (such as a read attribute response) shall indicate the TC is no longer
 1815 available. If the device fails to receive 3 APS encrypted data frames in a row it shall consider
 1816 the TC no longer accessible and initiate a search for it. Failure of the encryption or frame
 1817 counter shall constitute a failure of the keep-alive.

1818

1819 **5.4.2.2.3.5 Trust Center Swap-out Process**

1820 The following steps describe the Trust Center swap-out process.

1821 Preconditions: ESI installed and PAN formed.

- 1822 1 Back up Extended PAN ID to off-chip device (mandatory).
- 1823 2 SE device provisioned on the ESI and installed.
- 1824 3 Back up EUI64 and Install Code to off-chip device (optional).
- 1825 4 SE device performs CBKE successfully, derived TC link key and EUI64 backed up to off-chip
 1826 device (mandatory).

- 1827 **5** Any updates to the TC Link Key must be backed up to off-chip device (mandatory).
- 1828 **6** SE device sends periodic APS encrypted command to the TC.
- 1829 **7** ESI replaced with the Extended PAN ID, list of EUI64s and hashed TC link keys restored
1830 from backed up data. (Permit joining is not required to be on in the network). The TC link keys
1831 shall be treated as install code-derived link keys and unauthorized.
- 1832 **8** New TC forms a new network using new network key, new short PAN ID, and backup of
1833 extended PAN ID.
- 1834 **9** SE device detects TC no longer available (see sub-clause 5.4.2.2.3.4).
- 1835 a There are 4 possible cases at this point.
- 1836 i The TC is temporarily unavailable.
- 1837 ii The device missed a network key update.
- 1838 iii The TC changed channels to avoid congestion.
- 1839 iv The TC has been swapped out.
1840
- 1841 **10** SE device scans for the current Extended PAN ID (the short PAN ID will probably have
1842 changed).
- 1843 **11** Prior to performing a rejoin the device must backup in local storage its current TC link key,
1844 state of the link key (authorized or unauthorized), network key, and associated NWK and APS
1845 frame counters.
- 1846 **12** If the TC sends a Transport Key message encrypted using the device's existing TC link key and
1847 the device is able to successfully decrypt and rejoin the network, no further operations are
1848 necessary. The device can resume all normal operations.
- 1849 **13** If the TC sends a Transport Key message encrypted using a 128-bit AES-MMO hash of the TC
1850 link key, then the device shall ignore the frame counter check and accept the new network key.
1851 It will also record the source IEEE address of the sending device as the new TC address. It
1852 shall mark the hashed TC link key as not authorized and treat the link key as an installation
1853 code. It must now perform Key Establishment to fully authenticate itself in the network.
- 1854 **14** If the key establishment is NOT successful the device may try again immediately. Otherwise it
1855 shall leave that network. It can continue scanning for additional networks to rejoin, or restore
1856 the values of its previous network and resume normal operation.

1857 **15** If the Key Establishment is successful then the device can resume normal operation. It may
1858 discard the backup of security data from the old network. The TC must backup hash of the new
1859 TC Link Key for this device to an off-chip device.

1860 **16** If after attempting rejoin with all discovered PANs fails then the device shall fall back to the
1861 existing PAN.

1862 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.

1868 The hashed key shall be created by performing a 128-bit AES-MMO hash on the
1869 128-bit key data. The following is a test vector for the hash:

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

TC Link Key	C0C1C2C3C4C5C6C7C8C9CACBCCCDCECF
Hashed TC Link Key	A7977E88BC0B61E8210827109A228F2D

5.4.2.2.3.7 Trust Center (ESI/Meter)

1874 Dependencies

1875 The ESI shall support backup and restore of data (including TC link keys) to an off-chip
1876 device.

1877 Routers shall detect the TC is no longer available by sending an APS encrypted command and
1878 receiving the APS acknowledgement with a maximum periodic interval of 30¹ minutes.

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

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

¹ CCB 1746

1881 **5.4.3 Devices Leaving the Network**

1882 Upon receipt of an APS update device command indicating a device has left the network the
1883 trust center shall not remove the trust center link key assigned to that device. This is to prevent
1884 a device on the network performing a denial of service attack by spoofing the MAC address of
1885 another node and issuing a false ZigBee *Network Leave* command. Devices should be removed
1886 from Trust Center authorization and trust center link key lists via out of band methods, i.e. secure
1887 meter back haul or secure IP interface.

1888 Devices should follow the guidelines for stale keys described in 5.4.5.

1889 **5.4.4 Updating the Network Key**

1890 Periodically the trust center shall update the network key. This allows the trust center to phase
1891 out a previous instance of the network key so that devices that are no longer on the network will
1892 not be able to perform a secure rejoin. Those devices must then perform a rejoin, which
1893 allows the trust center to authorize whether or not they are allowed to be on the network.

1894 When the trust center wishes to update the network key it will broadcast the network key to
1895 all devices in the network. All devices receiving the key update will store but will not start
1896 using the new key.

1897 It is assumed that routers will receive the network key update sent by the Trust Center. Sleepy
1898 end devices are unlikely to get the network key update sent by the Trust Center unless the device
1899 polls frequently.

1900 After sending an updated network key, the trust center shall wait a minimum of
1901 nwkNetworkBroadcastDeliveryTime before sending the switch key message. Devices that miss
1902 the key switch broadcast message will implicitly switch when they receive any network
1903 message that is encrypted using the new key sequence number.

1904 Once the network has started using the new key, any device that has missed the key update
1905 message will not be able to communicate on the network. Those devices that missed the key
1906 update must follow the Re-joining a Secured Network procedure.

1907 **5.4.5 Updating the Link Key**

1908 Periodically the trust center may update the link key associated with a particular device. This
1909 allows the trust center to phase out the existing key and refresh it with a new key. The trust
1910 center can decide on its own what the policy is for how long a link key may be used and how
1911 often it should be updated.

1912 Trust Center link keys are used for sending application messages as well as stack commands.
1913 Therefore a trust center cannot simply delete a link key that it wants to update. The trust center
1914 must accept and or send encrypted APS commands to or from a device even if it has retired that
1915 link key from encryption of application data messages. This is especially necessary for sleeping

1916 end devices, which may not have the current network key and need to use their link key to
1917 obtain an updated copy during a rejoin.

1918 When the trust center deems that a particular link key should no longer be used, it shall mark the
1919 key as stale. A stale key shall not be used to send data messages. Devices that receive a
1920 message using a stale key should discard the message and shall not send an APS
1921 acknowledgement to the sender.

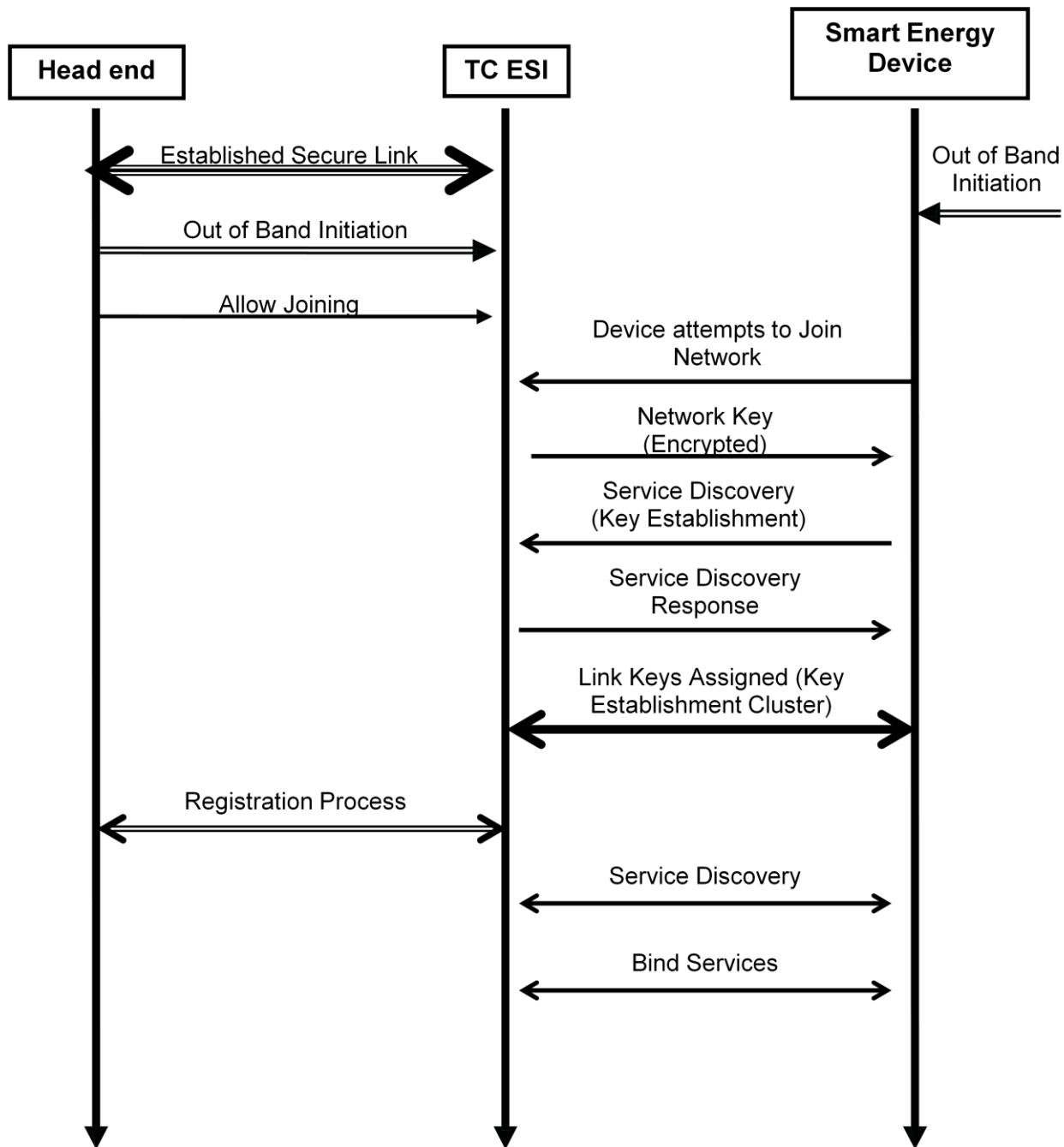
1922 Devices shall accept and process APS commands that are encrypted with a stale key.

1923 When the trust center receives a message encrypted with a stale link key, it shall initiate the key
1924 establishment procedure to negotiate a new link key. Upon successful establishment of the new
1925 link key with the device, the device shall clear the stale indicator for that key.

1926 Devices that are not acting as the trust center may utilize their own policy for retiring and
1927 updating application link keys with other devices that are not the trust center. Those devices are
1928 not required to keep around retired keys and therefore may delete them prior to establishing
1929 an updated link key using the Key Establishment cluster.

1930 [5.4.5.1 Network Joining and Registration Diagram](#)

1931 Figure 5-5 depicts an example of a successful network startup and certificate exchange (with
1932 pre-established link keys). Please refer to Annex C for further discussions on communication
1933 exchanges and key support.

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

1937 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 Key Establishment Cluster service discovery process whenever it detects the Smart Energy End Device fails to do so.

1942 After Joining and after Key Establishment:

- 1943 • A device supporting MDUs shall request MDU Pairing Information from the Trust Center (ESI). If MDU Pairing Information is returned, the device should limit service discovery to those devices identified in the returned information. If the device has joined a ‘normal’ Smart Energy HAN, no Pairing Information will be returned and the device should perform service discovery with all devices on the network. Refer to 5.6 and Annex D.13 for further details.
- 1949 • Client SHALL perform service discovery.
- 1950 • Sleepy devices SHALL perform “get” requests for data they wish to receive and SHOULD NOT expect to receive unsolicited messages.
- 1952 • If a Client wishes to receive unsolicited messages, Client SHALL follow with attempt(s) to ZDO Bind Request. A Client does not have to support a binding table.
- 1954 • If Server does not support binding, Server SHALL perform service discovery and register those devices for unsolicited messages (whether or not they want the messages).
- 1956 • For backward compatibility, Server SHOULD perform service discovery and register those devices for unsolicited messages (whether or not they want the messages).

1958

1959 **5.4.6 Cluster Usage of Security Keys**

1960 The SE Profile utilizes a higher level of security on the network but not all clusters need to utilize
 1961 Application Link keys. All clusters are required to use network layer encryption using the
 1962 network key. Table 5-13 identifies the security keys utilized by each cluster:

1963

1964

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

1965

1966

1967 Once a Registered SE device has an Application Link Key established with the ESI, it may also
 1968 establish Application Link Keys with any other device on the SE Network. This is accomplished by
 1969 using the ZigBee service and device discovery process (employing the Network Key). Note that,
 1970 in an MDU network, the service and device discovery process can only commence once MDU
 1971 Pairing information has been acquired, and will then be limited to devices within the same
 1972 ‘virtual HAN’. Regardless of the communication paths, all SE applications shall use and
 1973 validate the Security key usage as listed in Table 5-13. If link key encryption is NOT used but
 1974 required, the receiving device shall generate a ZCL Default Response, employing the Network
 1975 Key, with a FAILURE (0x01) status code.

1976 It is permissible for a device to initiate a ZCL exchange using an application link key even when not
 1977 required. If a device receives a message with link key security even though it is not required as per
 1978 Table 5-13, it shall accept the message. Additionally, if a response is sent then it shall use link key
 1979 encryption.

1980 Unless stated otherwise, any ZCL clusters added to a Smart Energy endpoint shall be APS
 1981 encrypted. ZCL clusters without APS encryption shall be located on another (non-Smart Energy)
 1982 endpoint. The same restrictions shall also apply to manufacturer-specific clusters.²

² CCB 1501

1983 5.4.7 Key Establishment Related Security Policies

1984 The following are the policies relating to Key Establishment that are recommended for Smart
1985 Energy networks.

1986

1987 5.4.7.1 Joining

1988 If the device does not need to perform discovery queries or other non-secure operations after
1989 it joins an SE network and receives the Network Key, it should immediately initiate Key
1990 Establishment with the Trust Center to obtain a new Trust Center Link Key.

1991 If Key Establishment fails with a result of UNKNOWN_ISSUER the device shall leave the
1992 network. A device that does not initiate Key Establishment with the Trust Center within a
1993 reasonable period of time MAY be told to leave depending on the network operator's policy. A
1994 maximum period of 20 minutes is recommended.

1995 Upon successful negotiation of a new Trust Center Link Key the device may communicate using
1996 clusters that require APS security.

1997

1998 5.4.7.2 Trust Center

1999 The Trust Center shall keep track of whether a particular device has negotiated a CBKE Trust
2000 Center Link Key, or whether only a preconfigured Trust Center Link Key exists. The Trust
2001 Center shall not use the preconfigured link key to send encrypted APS Data messages to the
2002 device. The Trust Center shall discard any APS encrypted APS Data messages that use the
2003 preconfigured link key, and it shall not send APS Acknowledgments for those messages.

2004 The Trust Center shall accept and send APS Data messages that do not use APS Encryption to a
2005 device that has not negotiated a CBKE Trust Center Link key provided that the security usage
2006 for that cluster allows using only Network layer security (encrypted with the Network Key). See
2007 sub-clause 5.4.6, "Cluster Usage of Security Keys".

2008 The Trust Center is required to be a Smart Energy device. It is required to support Key
2009 Establishment server on at least one endpoint, though it may support it on more than one
2010 endpoint. These endpoints shall be considered to all refer to the same logical ZigBee device
2011 type, in other words the Trust Center. Any negotiation or establishment of a link key on one
2012 endpoint applies globally to the Trust Center as a device and is not specific to an endpoint.

2013 The Trust Center shall have a means of adding and removing keys of specific devices that are
2014 part of the Smart Energy network. The specific means of doing this is outside the scope of this
2015 document.

2016 5.4.7.3 During Joining

2017 Normal operation of a device in a Smart Energy network requires use of a preconfigured
2018 link key, established by using the Installation Code (refer to sub-clause 5.4.6), to join a ZigBee

2019 Pro network. After joining the network a device is required to initiate key establishment using
2020 ECMQV key agreement with the Trust Center, to obtain a new link key authorized for use in
2021 application messages.

2022 Prior to updating the preconfigured link key using key establishment, the Trust Center shall
2023 not allow Smart Energy messages that require APS encryption. Although the node has a link
2024 key, that node has not been authenticated and thus the key's use is not authorized for application
2025 messages. Its use is still required for certain stack messages (e.g., the APS Command Update
2026 Device) and must be accepted by the trust center.

2027 In order to perform key establishment the device must discover an endpoint on the Trust Center
2028 that supports the Key Establishment server cluster. The joining device shall perform a ZDO
2029 Match Descriptor Request to determine what endpoint to use. This request shall be unicast to
2030 the Trust Center's short address of 0x0000.

2031 When a reply is received, it may contain multiple endpoints that indicate support for the Key
2032 Establishment server. The joining device may use any endpoint to perform key establishment.
2033 Link keys established using key establishment are global to the Trust Center device and are not
2034 specific to a particular endpoint.

2035 Once a node has been authenticated by the Trust Center and obtained an authorized link
2036 key using key establishment, it may communicate with the Trust Center using APS layer
2037 security. The Trust Center should accept valid APS encrypted message using that new link
2038 key. At this point the joining device can communicate to the Trust Center as a Smart Energy
2039 device.

2040 [5.4.7.4 After Joining](#)

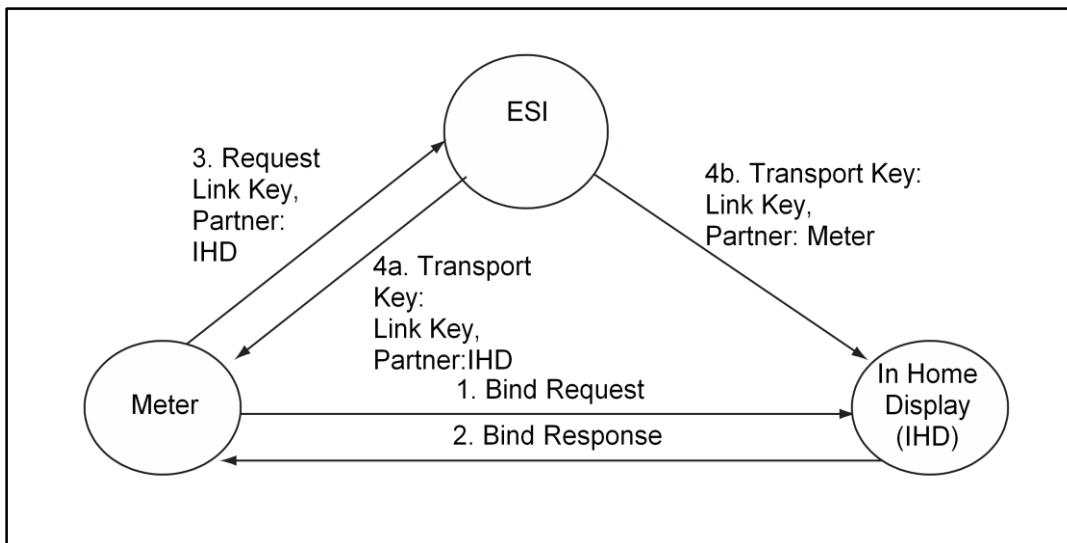
2041 After a node has joined, been authenticated using key establishment, and obtained an authorized
2042 link key, it may need to communicate with other nodes on the network using APS layer
2043 encryption.

2044 Rather than use key establishment with each node on the network, it would be advantageous to
2045 leverage the Trust Center to broker trust with other devices on the network. If two nodes have
2046 both obtained link keys with the Trust Center using key establishment, then they both trust the
2047 Trust Center. Both nodes will use the Trust Center to request a link key with each other. The
2048 trust center will respond to each node individually, sending a randomly generated link key. Each
2049 message will be encrypted using the individual nodes' link keys. The Trust Center would not
2050 send a link key to either node if one of the nodes has not authenticated using key establishment.

2051 The originating node would start this process by sending a bind request command with APS ack
2052 to the Key Establishment cluster of the destination device. If a bind confirm is received with a
2053 status of success, the initiating device will perform a request key of the trust center (for an
2054 application link key using the EUI of the other device in the pair). The trust center will then

2055 send a link key to each device using the key transport. If the bind confirm is received with a
 2056 status other than success, the request key should not be sent to the trust center.

2057 This functionality is optional, however support of this is required for ESI devices acting as
 2058 trust centers. All devices sending the request key command and the trust center should have a
 2059 timeout of 5 seconds.



2060

2061 **Figure 5-6 Node Communication with Other Nodes on the**
 2062 **Network Using APS Layer Encryption**

2063

2064 The advantages of using the stack primitives to request keys rather than key establishment
 2065 are that devices can forego the expensive ECC operations. Small microprocessors have
 2066 extremely limited resources and requiring full key establishment with all devices where link keys
 2067 are required is overly burdensome. In addition, ESIs may have other security policies in place
 2068 (such as node blacklists or certificate revocation lists) that individual nodes do not have
 2069 knowledge of, or have the resources to keep track of.

2070 Nodes that are not the trust center would not be allowed to initiate key establishment
 2071 with another device that is not the Trust Center. If a device receives an Initiate Key Establishment
 2072 Request from a device that is not the Trust Center, and it is not the Trust Center, it shall
 2073 terminate the key establishment immediately with a status of NO_RESOURCES. This ensures
 2074 that the ESI authenticates all devices with key establishment after joining, and limits the
 2075 use of key establishment in the network.

2076 Other ESI devices on the network that are not the trust center would have to go through the same
 2077 procedure as above, contacting the ESI trust center, in order to send/receive messages that require
 2078 APS layer encryption with another node.

2079 5.4.8 Security Best Practices

5.4.8.1 Out of Band Pre-Configured Link Key Process

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

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

- Step 1: An Installation Code is created and made available
 - Step 2: The Pre-configured Link Key is derived from the Installation Code using the Matyas-Meyer-Oseas hash function
 - Step 3: The Pre-configured Link Key is configured in the device

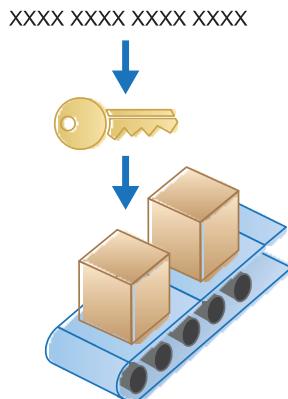
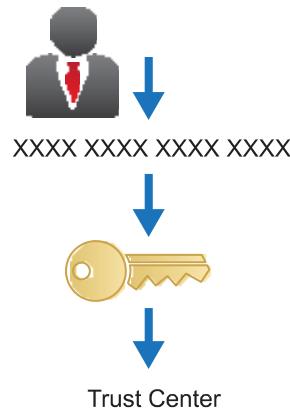


Figure 5-7 Smart Energy Device Installation Code Process

As portrayed in Figure 5-8, during the installation process the initial Trust Center Link Key is derived from the Installation Code and sent via an out of band communication channel to the Trust center (ESI). The Trust center uses this Key as the Trust Center Link Key to subsequently configure the Network Key of the associating device.

- 2118
 2119 Step 1: The Installation Code is sent out of band
 2120
 2121 Step 2: The Pre-configured Link Key is derived
 2122 from the Installation Code using the
 2123 Matyas-Meyer-Oseas hash function
 2124
 2125 Step 3: The Pre-configured Link Key is sent to
 2126 the Trust Center using the AMI network
 2127
 2128
 2129
 2130
 2131



- 2132
 2133 **Figure 5-8 Installation Code Use with the Trust Center**
 2134

2135 **5.4.8.1.1 Installation Code Format**

2136 The Installation Code consists of a 48, 64, 96, or 128 bit number and a 16 bit CRC (using CCITT
 2137 CRC standard polynomial $X^{16} + X^{12} + X^5 + 1$). When printed or displayed, Installation Codes
 2138 are represented as multiple groups of 4 hexadecimal digits.

2139 48 Bit example:

2140 Installation Code of “83FE D340 7A93 2B70”

2141 Where values 0x83, 0x FE, 0xD3, 0x40, 0x 7A, and 0x93 are used to calculate the CRC16
 2142 with the result returning 0x702B.

2144 *Note: The Octet order of the CRC code in the printed Installation code is Least Significant
 2145 Octet followed by Most Significant Octet, giving the printed result of “2B70”.*

2146 64 Bit example:

2148 Installation Code of “83FE D340 7A93 9738 C552”

2149 Where values 0x83, 0x FE, 0xD3, 0x40, 0x 7A, 0x93, 0x 97, and 0x38 are used to calculate the
 2150 CRC16 with the result returning 0x52C5.

2151

2152 96 Bit example:

2153 Installation Code of “83FE D340 7A93 9723 A5C6 39FF 4C12”

2154 Where values 0x83, 0x FE, 0xD3, 0x40, 0x 7A, 0x93, 0x 97, 0x23, 0xA5, 0xC6, 0x39 and 0xFF
 2155 are used to calculate the CRC16 with the result returning 0x124C.

2156

2157

2158

2159 128 Bit example:

2160 Installation Code of “83FE D340 7A93 9723 A5C6 39B2 6916 D505 C3B5”

2161 Where values 0x83, 0xFE, 0xD3, 0x40, 0x7A, 0x93, 0x97, 0x23, 0xA5, 0xC6, 0x39,
2162 0xB2, 0x69, 0x16, 0xD5, and 0x05 are used to calculate the CRC16 with the result
2163 returning 0xB5C3.

2164

2165 **5.4.8.1.1.1 CRC Algorithm Information**

2166 As stated earlier, the Installation Code CRC calculation is based upon the CRC
2167 16-CCITT algorithm and uses the following parameters:

2168 Length: 16
2169 Polynomial: $x^{16} + x^{12} + x^5 + 1$ (0x1021)
2170 Initialization method: Direct
2171 Initialization value: 0xFFFF
2172 Final XOR value: 0xFFFF
2173 Reflected In: True
2174 Reflected Out: True

2175 Open source implementations of the CRC 16-CCITT algorithm are available on the internet at
2176 sites like SourceForge and others. The source code is also available for download from the
2177 ZigBee document management system [B6].
2178

2179 **5.4.8.1.2 Hashing Function**

2180 An AES-128 key is derived from the Installation Code using the Matyas-Meyer- Oseas (MMO)
2181 hash function (specified in Annex B.6 in ZigBee Document 053474r17, The ZigBee
2182 Specification, ZigBee Technical Steering Committee (TSC) with a digest size (hashlen) equal to
2183 128 bits).

2184 Installation Code examples:

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

2187 *Note: Least significant byte is 0x83 and Most significant byte is 0x93.*

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

2188 *Note: Least significant byte is 0x83 and Most significant byte is 0x38.*

2189

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

2190 *Note: Least significant byte is 0x83 and Most significant byte is 0xFF.*

2191

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

2192 *Note: Least significant byte is 0x83 and Most significant byte is 0x05.*

2193

2194

2195

2196

2197

2198

2199 [5.4.8.1.2.1 MMO Hash Code Example](#)

2200 Open source implementations of the MMO Hash based on the Rijndael implementation
2201 are available on the internet at sites like SourceForge and others. The source code is also
2202 available for download from the ZigBee document management system [B6].
2203

2204 [5.4.8.2 Multiple Security Credentials](#)

2205
2206 The Key Establishment cluster supports multiple cryptographic suites and, as a result, a device's
2207 security credentials (e.g. certificates) are specific to the cryptographic suite. It is recommended
2208 that all devices (both joining and trust centers) contain credentials for all supported cryptographic
2209 suites to ensure interoperability in all networks where it may be deployed.
2210

2211 However, the trust center dictates security policy for the network and as such may require a
2212 higher level of security by using certain cryptographic suites as determined by the network
2213 owner. Therefore it is possible that trust center devices deployed into new regions with no
2214 existing Smart Energy may utilize only newer cryptographic suites that have a higher level of
2215 cryptographic strength.

2216 [5.5 Commissioning](#)

2217 Many, if not all of the devices described in this document, will require some form of
2218 commissioning, even if the user or installer doesn't see it. This is because, for example, a load
2219 control device needs to be bound to some sort of control device in order to perform its function
2220 and, even if the required initializations are done at the factory before the device is installed, the
2221 required operations are virtually the same as is the outcome.

2222 The ZigBee Alliance has recognized the importance of commissioning and, in particular, the
2223 importance of specifications for network and stack commissioning in a multi-vendor
2224 environment. Thus, network and stack commissioning procedures are being designed outside the
2225 context of any particular profile, where possible, and grouped under the auspices of the
2226 Commissioning Tools Task Group (CTTG). This task group is developing a commissioning
2227 framework specification [B2].
2228

2229 [5.5.1 Forming the Network \(Start-up Sequence\)](#)

2230 Smart Energy devices must form their own network or join an existing network. The
2231 commissioning framework [B2] discusses some of the relevant issues in this procedure.

2232 It is intended that an installer of a Smart Energy device knows if the device is forming a
2233 network or joining an existing network.

2234 If a device is forming a network there is no user interaction required since the form process
2235 can be completed by the device. However there should be some indication to the user or
2236 installer that the network has formed properly. The indication can be implemented in a
2237 number of ways including blinking indicator lights, colored indicator lights, arrays of indicator
2238 lights, text displays, graphic displays, audible indicators such as buzzers and speakers, or
2239 through separate means.

2240 If a device is joining an existing network, it will join the network using the processes
2241 outlined in sub-clause 5.4. Permit joining will have been turned on due to either installer action
2242 or some backchannel mechanism because of user or installer action. It is recommended there
2243 be some indication to the user that the device has joined the network successfully. The
2244 indication can be implemented in a number of ways including blinking indicator lights, colored
2245 indicator lights, arrays of indicator lights, text displays, graphic displays, audible indicators
2246 such as buzzers and speakers, etc.
2247

2248 **5.5.2 Support for Commissioning Modes**

2249 Three different commissioning modes are discussed in [B2]. They are denoted A, E and S-mode.

2250 As discussed above, Smart Energy devices will either automatically form or join a network based
2251 on the processes outlined in sub-clause 5.4.

2252 The pre-installation of start up parameters could be done at manufacturing (which is defined as A
2253 mode), by an installer tool at the dispatching warehouse, or on site (which would then be S
2254 mode). Devices that support this pre-installation must document the methods used for this
2255 preinstallation of parameters to accomplish this process.

2256 Those devices that will join an existing network must support button pushes or simple
2257 documented user interfaces to initiate the joining process. This is in support of E mode
2258 commissioning.
2259

2260 **5.5.3 Commissioning Documentation Best Practices**

2261 To ensure a uniform user experience when commissioning Smart Energy devices, all ZigBee
2262 Smart Energy devices are required to provide documentation with their product that explains how
2263 to perform device commissioning in using a common language set, i.e., “form network”, “join
2264 network”, etc. Please refer to [B2] for further guidance using installation tools and procedures.
2265

2266 **5.5.4 Commissioning Procedure for Different Network Types**

2267 Depending on the type of network being installed, the commissioning procedures may be
2268 slightly different. To ensure interoperability even within these different methods the specific
2269 steps are detailed here.

2270

2271 **5.5.4.1 Commissioning for Neighborhood Area Network or Sub-metering**2272
2273
2274
2275
2276
2277

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

2278
2279
2280
2281

- 1 There are two methods for joining such a device onto an existing network:
 - a The device is commissioned using a tool with the necessary network and security start up parameters to allow it to rejoin the network as a new device. The device can rejoin any device in the network since it has all the network information.
 - b The network has permit joining turned on by an external tool and the device joins this network and undergoes joining and authentication as any newly joined device.

2282
2283

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

2286
2287
2288
2289
2290

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

2291
2292**5.5.4.2 Commissioning for Home Area Network**2293
2294
2295
2296

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

2297
2298
2299
2300
2301

- 1 The Smart Energy network must be informed of the device that is to be joined. This is done through an out of band means which could include a web login, phone call to a service center, or handheld tool. Using this methodology the existing network is made aware of the device ID and security information appropriate for the device (per the Key Establishment Cluster described in Annex C).

2302
2303
2304

- 2 The Smart Energy network is put into permit joining ON for a period of time.
- 3 The installer/homeowner is prompted to press a button or complete a menu sequence that tells the device to attempt to join a network.

- 2305 **4** The device joins the network and is authenticated using the appropriate security mechanisms
2306 per the Key Establishment Cluster.
- 2307 **5** An indicator is provided for the installer/homeowner indicating the device has joined a
2308 network and authenticated properly or provides information about improper authentication.
- 2309 **6** The device can now operate normally on the network.

2310 **5.5.5 ZigBee Smart Energy Joining, Service Discovery, and Device Binding**
2311 **Requirements**

2312 Commissioning of a device into a ZigBee Smart Energy network should be easy, reliable, and
2313 deterministic. Ideally, a new device could be installed by the home owner or installer
2314 communicating the device install code out of band to the coordinator/trust center and then simply
2315 powering up the device or manually putting the device into a commissioning (auto-join) state.
2316 The device should automatically handle all the steps needed to discover and join the correct PAN
2317 and establish relationships with other devices in the HAN without user intervention. As network
2318 or HAN conditions change, the devices should be able to adapt automatically without user
2319 intervention. ZigBee Smart Energy networks are supposed to last for decades, but once
2320 commissioned, devices should require no user interaction in order to remain part of the ZigBee
2321 PAN.

2322 Devices that are configured with a Startup Parameter of two (un-commissioned) should
2323 automatically begin or make easily available a way to go to Auto-Joining State as described
2324 below. (See sub-clause 5.3.1 for the SE Profile Startup Parameter set.)

2325
2326 **5.5.5.1 PAN Auto-Joining State**

- 2327 **1** When auto-joining state is initiated, a device shall periodically scan all startup set channels for
2328 networks that are allowing joining. (See sub-clause 5.3.1 for startup set channel description). A
2329 recommended periodic schedule would be:
2330 a Immediately when auto-joining state is initiated.
2331 b If auto-joining state fails, retry once a minute for the next 15 minutes, jittered by +/- 15
2332 seconds.
2333 c If those joining states fail, then retry to join once an hour jittered +/- 30 minutes.
- 2334 **2** To find prospective networks to join, the joining node shall send Beacon Request packets on
2335 each channel, dwelling on each channel as specified by the ZigBee PRO specification beacon
2336 response window.
- 2337 **3** When a beacon is heard and it has the “Permit Joining” bit set, the device shall attempt to join
2338 that PAN. It is up to the implementation of the device to decide if it wants to survey all
2339 channels and build a list of joinable PANs before attempting a join procedure, or if it should

2340 attempt to join each PAN on a beacon-by-beacon basis. The device shall use its preconfigured
2341 link key (derived from a hash of the installation code) to join the targeted SE PAN.
2342 Exchanging keys in the clear or with well known preconfigured link keys is not allowed.

- 2343 **4** If the device joins the network but receives a network key that it cannot decrypt, then it has
2344 likely joined an incorrect PAN and should back out and try the next joinable PAN. This
2345 situation happens most often when the out of band mechanism to communicate the installation
2346 key is flawed, or when more than one PAN is allowing joining. It is permissible to try and join
2347 the same network again, but not recommended that it be done more than three times in
2348 succession. It is expressly not allowed that a device repeat this step more than ten times
2349 without backing off to step two and scanning for other networks to join.
- 2350 **5** After the device joins the PAN and is granted ZigBee network key, it must perform service
2351 discovery to find a ZigBee Key Establishment cluster server, then perform ZigBee key
2352 establishment in order to get an APS layer link key.
- 2353 **6** If this key establishment fails, it is likely that one side of the exchange is configured with an
2354 invalid certificate or with no certificate at all. It is permissible to retry this step multiple times
2355 in succession, but it is expressly not allowed that a device repeat this step more than ten times in
2356 succession without pausing for a minimum of least fifteen minutes. Since the device was able
2357 to get a network key from the Trust Center, the device must have found the correct PAN to
2358 join, so there is no need to leave the network. A device that does not initiate Key
2359 Establishment with the Trust Center within a reasonable period of time MAY be told to leave
2360 depending on the network operator's policy. A maximum period of 20 minutes is
2361 recommended.
- 2362 **7** Once key establishment succeeds, the device has joined the correct PAN and shall never leave
2363 the PAN without direction from another device in the network (typically an APS *Remove*
2364 *Device* command from an ESI or ZigBee Network Manager) or direction from the user via the
2365 device user interface. Example user interfaces could be a text menu or a simple button push
2366 sequence. It is strongly recommended that the user interface procedure to get a device to leave
2367 the PAN be explicit and difficult to trigger accidentally. Leave commands received over the air
2368 should only be followed if the command is an APS encrypted APS *Remove* command.
2369 Network layer leave commands should be ignored unless the device is an end device, and the
2370 network leave command originated from the parent device.
- 2371 **8** A device that leaves a ZigBee network shall discard its network settings and link key, and
2372 revert to its install code, and wait for user input to return or automatically return to auto-join
2373 state step one. The device will require the out- of-band registration process to join a new
2374 network.
- 2375

2376 5.5.5.2 Service Discovery State:

- 2377 1 After successfully performing key establishment, devices supporting MDU shall perform
2378 service discovery for the MDU Pairing cluster first and, if found, request pairing information
2379 before continuing discovery for other services. If the MDU Pairing cluster is NOT located on
2380 the Trust Center (ESI), the device providing the cluster MUST be available on the HAN
2381 before other devices are allowed to join. If MDU Pairing Information is returned, the device
2382 should limit service discovery to those devices identified in the returned information. If the
2383 device has joined a ‘normal’ Smart Energy HAN, no Pairing Information will be returned and
2384 the device should perform service discovery with all devices on the network. Refer to 5.6 and
2385 Annex D.13 for further details.
- 2386 2 The device should use ZigBee Service Discovery mechanisms to discover other devices on the
2387 network that have services that match with the device's. This would apply to ZigBee Smart
2388 Energy clusters that support asynchronous event commands, like DRLC, Messaging, and Price
2389 clusters. For example, a load control device would use ZigBee service discovery to find ESIs
2390 that support the load control cluster server. (See sub-clause 5.4.5.1 for more details.).
- 2391 3 Where a network supports multiple commodity types, the service discovery mechanism may
2392 find multiple instances of certain clusters. In this case, the device should refer to the
2393 *CommodityType* attribute within any Price clusters found, or the *MeteringDeviceType* attribute
2394 within any Metering cluster servers found, in order to determine the appropriate endpoint(s).
2395 Price clusters must always be used where multiple commodities are supported; all clusters
2396 related to the same commodity should be grouped on the same endpoint. If an endpoint
2397 supporting the required commodity is not found, a device shall not use clusters identified as
2398 being for another commodity; the device shall repeat the discovery mechanism at regular
2399 intervals until such time as the correct clusters become available (see point 6 below for
2400 suggested intervals). Clusters for multiple commodities shall not be mixed on the same
2401 endpoint (see section 6.3 for further details).
- 2402 4 When a matching service is discovered, the device shall use ZigBee device bind mechanisms
2403 to send a binding request to the matching device endpoint. It is possible that more than one
2404 device with matching services will be discovered. If the device is not an ESI and the ESI are
2405 the matching device(s), the device should send binding requests to all ESI with matching
2406 services. See the “Multiple ESI Application Guidelines” for more details. Hence a device that
2407 wishes to receive unsolicited messages from an ESI on the Messaging Cluster, Price Cluster,
2408 DRLC Cluster, shall issue a bind request to the ESI for each cluster it is interested in.
- 2409 5 A device that sends a binding request is simply announcing itself to an ESI that it desires
2410 certain sets of information that the ESI may presently have or may obtain in the future, such as
2411 pricing information or DRLC event schedules. The ESIs that receive bind requests are free to
2412 refuse them, but if they refuse the binding request, they must choose another method (an
2413 address table for instance) to note the device's interest. Once a device has issued the binding
2414 request, it does not need to receive a binding response success. If the device receives a

2415 NOT_SUPPORTED (or other non-success code) response to a cluster device bind request, it
2416 should still send binding requests for any remaining clusters that it has not sent already.

2417 **6** After the device has discovered and bound to matching services, it has now established an
2418 application layer relationship with all other relevant devices in the HAN (see sub-clause 5.7.2
2419 for details of how to deal with multiple time servers and other duplicated services). That does
2420 not mean that the HAN is static and will not acquire new devices, replace devices, or power on
2421 devices that were not present during the initial discovery phase. To account for a dynamic
2422 HAN, devices shall:

- 2423 a Repeat the discovery phase on a period of no more than once every three hours and no less
2424 than once every 24 hours.
- 2425 b Repeat the discovery phase after successfully exiting the Rejoin and Recovery Phase (see
2426 below).
- 2427 c Optional - Repeat the discovery phase when a device announce broadcast for a full function
2428 device is received. The beginning of the discovery phase should be jittered between 60 and
2429 600 seconds and should be directed only at the device that sent the device announce
2430 broadcast.

2431
2432 **5.5.5.3 Device Steady State**

2433 This is the normal state of the device.

2434 **1** A device should make efforts to remain on the correct channel of the PAN and also to keep its
2435 network and application keys in sync with the trust center. It is possible that the device has
2436 missed a key roll or a channel change due to interference or while it has been powered down
2437 or asleep. In order to detect these types of network changes devices shall perform some sort of
2438 APS layer message exchange with an ESI on a regular basis. This is to establish that the device
2439 can still communicate with the ESI using a current network and APS layer key. This exchange
2440 should be performed in accordance with the keep-alive method described in sub-clause
2441 5.4.2.2.3.4. Devices that do not support APS encrypted clusters (Range Extenders for example),
2442 do not need to send APS encrypted packets to the Trust Center, but can send network
2443 encrypted packets instead.

2444 **2** What periodic APS layer message exchange is performed is up to the implementation.
2445 Examples would include:

- 2446 a Reading a mandatory Time cluster attribute (such as *CurrentTime*) on the ESI
2447 (recommended). This should work for all ESI.
- 2448 b Reading the current consumption attribute on the Metering cluster (if the ESI supports the
2449 Metering server).
- 2450 c Requesting next pricing info from the ESI (if the ESI supports the Price cluster server).

- 2451 **3** If the device attempts to perform the periodic message exchange and it fails for any reason, the
2452 device should note the failure and retry another exchange later. If after no more than twenty-
2453 four hours of retries have failed, the device shall go into the Rejoin and Recovery Phase. It is
2454 left to the implementation to decide how many retries should occur within the 24 hour period.
2455 It is also permissible for the device to enter the Rejoin and Recovery Phase earlier than 24
2456 hours based on number of failed retries or other factors.
- 2457 **4** Sleepy end devices are not required to periodically communicate with an ESI. Instead they
2458 should periodically poll their parents and if no parent is found after a suitable period find and
2459 rejoin to a new parent. If no parent is found on the original channel, the end device should
2460 enter the Rejoin and Recovery phase described below to find a new parent.

2461 **5.5.5.4 Rejoin and Recovery State**

2462 A device in Rejoin and Recovery Phase is trying to get in sync with its PAN.

- 2463 **1** The device in R&R Phase shall first attempt a ZigBee secure rejoin procedure on its current
2464 channel. If the secure rejoin procedure succeeds, the device should revert to its steady state
2465 behavior.
- 2466 **2** If the secure rejoin procedure fails, it shall attempt to do a trust center rejoin procedure on its
2467 current channel.
- 2468 **3** If the trust center rejoin procedure fails, it may optionally retry steps one and two up to three
2469 times.
- 2470 **4** If all attempts to rejoin on the current channel fail, the device shall scan all other channels for
2471 its PAN by issuing beacon requests. Note that the PAN ID may have changed and the device
2472 shall compare with the extended PAN ID in the beacon and not the short PAN ID.
- 2473 **5** If the device finds an extended PAN ID match in a received beacon, it shall repeat steps one
2474 and two on the new channel.
- 2475 **6** If the rejoin (and optional retries) fail on the new channel, the device shall continue scanning
2476 all remaining channels for its PAN.
- 2477 **7** If no correct PANs are discovered on any channel, the device shall return to its original
2478 channel to wait for the next R&R attempt.
- 2479 **8** If all rejoin attempts on all channels fail, the device shall return to its original channel to wait
2480 for the next R&R attempt. This means that the device is back on the original PAN channel, is
2481 still a member of the original PAN, (it has not left the network, and has not discarded any PAN
2482 information or security keys), and is simply waiting for the rest of the PAN to appear or to time
2483 out and begin another R&R attempt.

2485 **9** If while waiting for the next R&R attempt, the device receives an APS encrypted message
2486 from an ESI and is encrypted with the device's current network and APS layer key, the device
2487 shall leave the R&R phase and proceed to the steady state phase.

2488 **10** While in the R&R phase, the device shall retry steps 1-8 periodically, at least once every 24
2489 hours. Sleepy end devices may use a longer period. After four failed rejoin attempts, devices
2490 should not try to rejoin any faster than once per hour, with a jitter of +/- 30 minutes.

2491 **5.5.5.5 ESI Specific Considerations**

2492 **1** ESI that are not the PAN coordinator, trust center, or network manager shall perform the
2493 steady state phase and rejoin and recovery phase as described above.

2494 **2** ESI shall support at a minimum, through bindings or other means, at least five separate
2495 devices, with enough resources for each device to bind to all of the relevant clusters that the
2496 devices may request bindings to. For example, if the ESI supports five smart energy clusters
2497 that devices may send binding requests for, the ESI must support twenty five binding
2498 relationships, as well as five sets of device ids and security keys

2499 **3** It is strongly recommended that ESI operators remove inactive or deprecated devices from the
2500 HAN as well as ESI key and binding tables before adding new devices in order to make room
2501 for the new device(s). This use case is an example of a device replacement in the HAN.

2502 **4** The Trust Center shall never issue an APS *Remove* command without an explicit request from
2503 another device on the network or from the head-end network management system.

2504 **5** When a new device is registered with the ESI, the ESI may not have enough resources to
2505 support it. If the ESI is low on resources, it should notify the installer or ESI administrator (this
2506 could be via user interface, or backhaul/ backchannel communication for example.) The ESI
2507 shall not automatically remove other devices in order to free up resources for the new device
2508 without explicit approval from the installer or ESI administrator.

2509 **6** If a device joins the PAN, but does not successfully perform Key Establishment, a trust center
2510 may remove the device. This shall only be done after more than 1 hour has elapsed since the
2511 device's initial join. This shall be sent directly to the router, or to the parent of an end device. A
2512 child that receives a NWK leave from its parent when it does not have an authorized link key
2513 (i.e. not performed key establishment successfully) shall not ignore the leave.

2514 **7** It is permissible and encouraged that an ESI perform its own service discovery procedure after
2515 power up and on a periodic basis. The ESI may independently create its own bindings to
2516 devices with matching services. This may in some cases establish application layer
2517 relationships faster than waiting for devices to request bindings by themselves. As specified in
2518 sub-clause 5.4.5.1, an ESI that does not support bindings shall perform its own service
2519 discovery.

2520

2521 **5.6 Federated Trust Center Application Guidelines**

2522 When considering Smart Energy systems to support multiple dwelling units (MDUs) such as
2523 apartment blocks, one solution is to use a single ZigBee mesh network (NAN) to support the
2524 entire building. In this scenario, there will be a single Trust Center (ESI), also known as a
2525 *Federated Trust Center*, that will control this entire network.

2526 The devices associated with a particular apartment within the MDU will form a ‘virtual HAN’,
2527 and will consist of devices within the apartment together with the associated meter(s) which may
2528 be located in a central meter room or elsewhere within the building. Thus the overall network for
2529 the entire building will be made up of a number of groups of devices.

2530 The Federated Trust Center will need to be pre-loaded with information indicating the devices
2531 that will form the ‘virtual HAN’ for each apartment. Additional functionality will be required so
2532 that a device joining this NAN can be made aware of the other devices that will constitute the
2533 virtual network for the relevant apartment; the device will then limit its subsequent service
2534 discovery to that list of devices.

2535 Once a device has joined the NAN, and has successfully completed key establishment with the
2536 Trust Center, it shall request Pairing Information from the Trust Center. Once Pairing
2537 Information is returned, the device shall limit service discovery to those devices identified in the
2538 response. It should be noted that, in this scenario, service discovery will be performed using
2539 unicast commands rather than broadcast.

2540 Devices may join and leave the ‘virtual HAN’ from time to time. To allow for this possible
2541 reorganization, each device within the NAN shall periodically ask the Federated Trust Center
2542 whether new Pairing Information is available. The typical period is once every 24 hours. A
2543 *Version* field within the relevant commands allows updated information to be identified.

2544 For further information, refer to the MDU Pairing cluster definition in Annex D.13.

2545 **5.7 Multiple ESI Application Guidelines**

2546 **5.7.1 Overview**

2547 The ZigBee Smart Energy Profile allows for the use of multiple ESIs in a HAN. This feature is
2548 desirable from a reliability perspective, plus opens opportunities for vendors to innovate and
2549 provide additional services and functionality. Multiple ESIs does not mean multiple Trust Centers,
2550 only a single Trust Center is supported in a HAN.

2551 Clients may assume that all SE messages/directives (Demand Response events, price
2552 publishing, messaging) are created by the same entity, e.g., utility or energy management entity,

2553 or set of coordinated entities. These messages can be sent to devices via one or more transport
2554 mechanisms (in the HAN, this means the same message may be sent from multiple ESIs). A
2555 message with a specific ID typically will be unique within the system, even though a device may
2556 receive this message more than once. However, in a HAN with multiple, uncoordinated
2557 commodity service providers (e.g., gas vs. water, household electricity vs. PEV electricity), there
2558 is a possibility that different, unique events will have conflicting event IDs. Since it is expected
2559 that ID conflicts for events occurring at similar points of time will be rare, clients may ignore the
2560 issue and always assume that conflicting event IDs are duplicates. More complex clients may
2561 choose to better track events by service provider, commodity type, etc.

2562 *Note: The functionality in this section does not describe the operation of multiple ESIs within a*
2563 *MDU/NAN. Refer to section 5.6 for details of Federated Trust Center functionality.*

2564 **5.7.2 Device Behavior**

2565 **5.7.2.1 Service Discovery in Multi ESI Environments**

2566 A device should make itself aware of any and all ESIs in a SE HAN using service discovery. It
2567 shall perform this service discovery upon joining a network, power up (and network rejoin), and
2568 periodically. The typical period is once every 24 hours. A device that discovers an ESI with
2569 matching services shall create bindings on the ESI so that the ESI will register the device and
2570 send it appropriate SE commands. Devices which do not bind in a multiple ESI network are
2571 expected to poll the ESIs. ESIs that are not rediscovered over the period of multiple
2572 discovery cycles may be forgotten by the device.

2573 A device that discovers more than one ESI should determine a single ESI as an authoritative
2574 time source. To do so, it should use the Time cluster Master, Synchronized, and Superseding bits.

2575 A Time server with the Superseding bit set will always take precedence over a Time server
2576 without that bit set, including ones that have the Master bit set. A new ESI going into a faulty
2577 installation can set the Superseding bit and take over the network's Time synchronization.
2578 However, it is not required for SE 1.x to have this bit set if the new ESI does not want to
2579 forcefully take over the Time server role. This bit is set independently of the other three
2580 *TimeStatus* bits (Master, Synchronized, MasterZoneDst).

2581 **5.7.2.2 Determining the Most Authoritative Time Source**

2583 Devices shall synchronize to a Time server with the highest rank according to the following
2584 rules, listed in order of precedence:

- 2585 1 A server with the Superseding and Master bits set shall be chosen over a server with only the
2586 Master bit set.
- 2587 2 A server with the Master bit set shall be chosen over a server without the bit set.

2588 **3** The server with the lower short address shall be chosen (note that this means a coordinator
2589 with the Superseding and Master bit set will always be chosen as the network time server).

2590 **4** A Time server with neither the Master nor Synchronized bits set should not be chosen as the
2591 network time server.

2592 [**5.7.2.3 Periodic Time Source Checking During Normal Operation**](#)

2593 During normal operation (the most authoritative time source is found and it has valid time),
2594 clients periodically repeat the time source scan to pick up new, more authoritative time sources,
2595 as per the following rules:

2596 **1** Non-sleepy clients shall locate the most authoritative time source at least once every 24 hours.

2597 **2** Sleepy devices should locate the most authoritative time source at least once every 24 hours.

2598 **3** Clients shall scan for time sources after rebooting, and after joining or rejoining the network.

2599 [**5.7.2.4 Invalid Time and Interim Time Sources**](#)

2600 Although the rules above are used to find the most authoritative time source for the network,
2601 there are conditions where what would normally be the most authoritative source is
2602 temporarily unable to provide valid time. In this situation, regardless of whether it is encountered
2603 as part of the original time discovery or the periodic rediscovery, devices obey the following
2604 rules:

2605 **1** If a server is temporarily unable to provide valid UTC, it shall report all time attributes (e.g.,
2606 UTC, local time) as 0xFFFFFFFF. It should leave the Superseding, Master, and Synchronized
2607 bits set as if it did have valid time.

2608 **2** If the most authoritative time source for the network has invalid time, clients should
2609 temporarily use the Time server of next highest rank, but shall periodically look for the more
2610 authoritative server(s) to obtain valid time. Non-sleepy devices shall check at least once every
2611 15 minutes. Sleepy devices may check as often as their power budget will allow.

2612 **3** When a more authoritative time source with valid time is found, clients shall immediately
2613 switch to using that source's time basis.

2614 [**5.7.2.5 Handling SE Commands from Multiple ESIs**](#)

2615 When a device creates bindings on multiple ESIs, it may receive SE commands from those
2616 ESIs. Simple device logic such as assuming all commands came from the PAN coordinator is not
2617 appropriate. The following rules describe the desirable device behavior.

- 2618 1 When a device receives an event (Demand Response, Price, Messaging) any time reference in
2619 the message should be viewed in context with the time reference of the most authoritative ESI
2620 time server.
- 2621 2 When a device receives duplicate events (same event ID) from multiple ESIs, it shall send an
2622 event response to each ESI. Future duplicate events from the same ESI(s) shall be either
2623 “ignored” by sending no response at all or with a default response containing a success status
2624 code.
- 2625 3 Conflicting events with the same event ID from different ESIs will be resolved by the device in
2626 the same manner as if they came from a single ESI.
- 2627 4 When a device has an asynchronous follow up event response it should send the response to
2628 the ESIs that created the condition. If the event was received from more than one ESI, the
2629 device shall send the asynchronous event response to all ESIs from which it received the event.

2630 **5.7.2.6 Handling Multiple Uncoordinated Back-end Systems**

2631 When multiple, uncoordinated service providers deploy ESIs in a HAN, it is possible that
2632 different back-ends/ESIs will have a different notion of time. However, only one ESI will
2633 be the authoritative time source for the HAN. In this scenario, an ESI may require a mechanism
2634 to ensure that its events are executed on its time basis, even if it is not the authoritative time
2635 source. ESIs that require this behavior may make use of the following application guidelines:

- 2636 1 An ESI may implement the Time cluster client and determine the most authoritative time
2637 source for the HAN, using the rules defined above.
- 2638 2 If the ESI is not the authoritative time source, it may synchronize its clock, or apply
2639 differentials to the start time of its events, to ensure that clients execute those events on the
2640 intended schedule.
- 2641 3 If the more authoritative time server disappears and is not seen for 24 hours, clients may
2642 assume that the server has left the network, and resume normal operation using the most
2643 authoritative time server (with valid time) that remains.

2644 **5.8 Other Smart Energy Profile Requirements and Best Practices**

2645 **5.8.1 Preferred Channel Usage**

2646 When forming a new network, or scanning to join a network, Smart Energy devices should
2647 do channel scans using the following preferred channels before scanning the rest of the
2648 channels in order to avoid the most commonly used WiFi channels. This is to improve the user
2649 experience during installation (quicker joining) and possibly improve bandwidth (on average).

2650 **Preferred 2.4 GHz Channels - 11, 14, 15, 19, 20, 24, 25**

2651 **Preferred 900MHz Channels** – Use all available for ZigBee.

2652 **5.8.2 Broadcast Policy**

2653 Broadcasts are strongly discouraged for Smart Energy devices. Devices are limited to a
2654 maximum broadcast frequency of one broadcast per second and strongly encouraged to exercise
2655 broadcasts much less frequently.

2656 **5.8.3 Frequency Agility**

2657 Frequency Agility would only be officially exercised in a network by a system controller, or
2658 higher functioning device (ESI, aggregator, installation tool, etc...). Devices may support
2659 frequency agility hooks to be commanded to “go to channel X”. Devices that do not support
2660 frequency agility may implement either the NWK rejoin or orphan join feature to find a network
2661 that has changed channels.

2662 **5.8.4 Key Updates**

2663 Energy devices are only required to support ZigBee “residential mode” security or ZigBee
2664 PRO “standard mode” with the required use of link keys. All link key updates shall use the
2665 Key Establishment Cluster. Sleeping devices that miss key updates can request a new key using
2666 the existing link key so there is no problem with sleeping devices missing key updates.

2667 DELETED SECTION 5.8.5

2668 **5.9 Coexistence and Interoperability with HA Devices**

2669 It is desirable to allow interoperability of HA and Smart Energy devices where practical.
2670 However, it is undesirable to publicly share keys during the joining process or share private
2671 information over a less secure network. HA devices that only provide functionality for
2672 receiving network keys in the clear during a join process cannot be used in a Smart Energy
2673 network. HA devices may also be extended with Smart Energy clusters providing they support
2674 the use of Link Keys and the Smart Energy security models. If so, they can be certified as
2675 HA and Smart Energy capable allowing those devices to operate either in an HA network or a
2676 Smart Energy network.

2677 **5.10 Device Descriptions**

2678 Device descriptions specified in this profile are summarized in Table 5-14 along with their
2679 respective Device IDs. The devices are organized according to the end application areas they
2680 address. A product that conforms to this specification shall implement at least one of these
2681 device descriptions and shall also include the device descriptions corresponding to all
2682 applications implemented on the product where a standard device description is specified in this
2683 profile. For example, if a product implements both a thermostat and an In-Home Display, then
2684 the thermostat and In-Home Display device descriptions must both be supported.

2685 This list will be added to in future versions of the profile as new clusters are developed to meet
 2686 the needs of manufacturers. The reserved values shall not be used until the profile defines them.
 2687 Manufacturer-specific device descriptions shall reside on a separate endpoint and use a private
 2688 profile ID.

2689 **Table 5-14 – Devices Specified in the Smart Energy Profile**

	Device	Device ID
Generic	Range Extender	0x0008
Smart Energy	Energy Service Interface	0x0500
	Metering Device	0x0501
	In-Home Display	0x0502
	Programmable Communicating Thermostat	0x0503
	Load Control Device	0x0504
	Smart Appliance	0x0505
	Prepayment Terminal	0x0506
	Physical Device	0x0507
	Remote Communications Device	0x0508
	Reserved	0x0509 – 0x5FF

2691

2692 **5.11 ZigBee Cluster Library (ZCL)**

2693 This profile utilizes some of the clusters specified in the ZigBee Cluster Library. The implementation details for each cluster are given in the ZCL specifications. Further specification and clarification is given in this profile where necessary.

2694 The ZCL provides a mechanism for clusters to report changes to the value of various attributes. It also provides commands to configure the reporting parameters. Products shall support the attribute reporting mechanism for supported attributes as specified in the ZCL. The minimum reporting interval specified in the ZCL [B1] shall be set to a value greater than or

2701 equal to 0x0001. The maximum reporting interval should be set to 0x0000 by default, and if it is
2702 set to a non-zero value it shall be set to a value greater than or equal to 0x003C and greater than
2703 the value of the minimum reporting interval. These settings will restrict the attributes from
2704 being reported more often than once every second if the attribute is changing quickly and at least
2705 once every minute if the attribute does not change for a long time. It is recommended that the
2706 minimum reporting interval be set to one minute and the maximum reporting interval be set to a
2707 much greater value to avoid unnecessary traffic.

2708 Devices shall use the ZCL default response error handing. Typical examples of this are:

- 2709 • When receiving commands that don't have data collected such as Get Scheduled Events, Get
2710 Current Price, Get Scheduled Prices, Get Block Period(s), and Get Last Message, devices
2711 shall respond using the ZCL default response with a status code of NOT_FOUND.
- 2712 • When receiving requests for unsupported commands, devices shall respond using the ZCL
2713 default response with a status code of UNSUP_CLUSTER_COMMAND.
- 2714 • When receiving malformed commands, devices shall respond using the ZCL default
2715 response with a status code of MALFORMED_COMMAND.
- 2716 • When receiving requests for accessing unsupported attributes, devices shall respond using
2717 the ZCL default response with a status code of UNSUPPORTED_ATTRIBUTE.
- 2718 • When receiving requests for accessing attributes where there is an additional access control
2719 mechanism and access is denied, devices shall respond using the ZCL default response with
2720 a status code of NOT_AUTHORIZED. Any additional access control mechanism is optional
2721 and the specification of the additional access control mechanism is out of scope of this
2722 specification.

2723 Please refer to [B1] for additional status codes support in the ZCL default response.

2724 **5.12 Cluster List and IDs**

2725 The clusters used in this profile are listed in Table 5.15. The clusters are listed according to
2726 the functional domain they belong to in the ZCL and indicate the additional new Smart
2727 Energy clusters. The existing corresponding ZCL General cluster identifiers can be found in the
2728 ZCL [B1].

2729 The functionality made available by all supported clusters shall be that given in their ZCL
2730 specifications except where a device description in this profile includes further specification,
2731 clarification or restriction as needed for a particular device.

2732 Most clusters include optional attributes. The application designer must be aware that optional
 2733 attributes might not be implemented on a particular device. All Smart Energy devices must
 2734 discover and deal with unsupported attributes on other devices.

2735 It is expected that clusters will continue to be developed in the ZCL that will be useful in this
 2736 profile. In many cases, new clusters will be organized into new device descriptions that are
 2737 separate from those currently defined. There may also be situations where it makes sense to add
 2738 clusters as optional elements of existing device descriptions.

2739 Manufacturer-specific clusters may be added to any device description in this profile as long
 2740 as they follow the specifications given in the ZCL [B1].

2741

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

2742

2743

2744 **5.12.1 ZCL General Clusters**

2745 Except for the Key Establishment Cluster, which is covered in Annex C, please refer to the
2746 ZCL Cluster Specification [B1] for the General Cluster descriptions.

2747 **5.12.1.1 ZCL Time Cluster and Time Synchronization**

2748 The Smart Energy profile requires time synchronization between devices to properly
2749 support the coordination of Demand Response/Load Control events, Price changes, and the
2750 collection of metered data. In order to simplify the understanding of time, the Smart
2751 Energy profile will leverage UTC as the common time base. To this end a new ZCL
2752 attribute data type, UTCTime is included and its definition can be found in Annex A.

2753 It is desired for the processes for synchronizing time to be as network friendly as possible to
2754 eliminate excessive traffic. To support this, time accuracy on Client devices shall be within +/-
2755 1 minute of the server device (ESI) per 24 hour period. The Client devices shall design a
2756 clock accuracy that never requires more than one time synchronization event per 24 hour period.
2757 The exception to this is when devices need to rejoin or re-register on the network. Again, the
2758 desire is to keep time synchronization traffic to a minimum.

2759 Further, implementers must be aware that network communication delays will cause minor
2760 differences in time between devices. The Smart Energy profile expectations are that this will
2761 be a minor issue given the use cases it's fulfilling. It will not nor does it recommend
2762 implementers develop an NTP or equivalent scheme to compensate for network delays. These
2763 methods are viewed as having the potential to cause excessive network communications.

2764 **5.12.1.2 Transaction Sequence Numbers**

2765 The normal usage of Transaction Sequence Numbers is defined in [B1] 2.3.1.3. The following
2766 clarifications shall be noted with respect to the usage of Transaction Sequence Numbers within
2767 the Smart Energy Standard:

2768 1 Where a ‘Publish’ command is sent in response to the receipt of an associated ‘Get’
2769 command, the ‘Publish’ command shall use the Transaction Sequence Number received in
2770 the ‘Get’ command

2771 2 Where a ‘Publish’ command is sent unsolicited, the server is free to choose any value of
2772 Transaction Sequence Number

2773 A Transaction Sequence Number is not to be used to reject a command, only to correlate
2774 messages.

2775

2776 **5.13 Coexistence with devices using other Profiles**

2777 Devices that do not implement Smart Energy 1.x clusters may be allowed on to the ZigBee
2778 network at the discretion of the network owner. Additional router devices have the benefit of
2779 growing the range and reliability of the ZigBee network, and offering additional services not
2780 offered by the Smart Energy profile. This also benefits the user as it means fewer ZigBee
2781 networks to manage.

2782 This section describes only how Smart Energy devices and devices from other profiles will
2783 interact in a general sense to join the network and communicate. It does not describe how
2784 specific clusters from one profile may be used to communicate with clusters on a different
2785 profile.

2786 All devices in the Smart Energy protocol are required to adhere to these requirements regardless
2787 of whether the network owner chooses to allow devices from other profiles on their network.

2788 **5.13.1 Requirements for the Smart Energy Trust Center**

- 2789 1 The Trust Center shall only allow devices onto the Smart Energy network that utilize
2790 installation code based link keys. This is the normal behavior per the Smart Energy profile.
- 2791 2 If a Trust Center does not want to allow non-Smart Energy devices onto the network, it is
2792 recommended that the TC should check whether a joining device supports the Key
2793 Establishment cluster and, if not, request the joining device to leave the network.
- 2794 3 It is recommended that Smart Energy devices that fail CBKE should remove themselves
2795 from the network.
- 2796 4 Trust Centers that support both Smart Energy and non-Smart Energy devices should not
2797 remove non-Smart Energy devices that do not attempt CBKE.
- 2798 5 If the Trust Center sends NWK key updates via unicast, then it shall still include devices
2799 from other profiles that have not performed CBKE.
- 2800 6 Per normal requirements, the Trust Center shall not grant partner link keys to devices that
2801 have not performed CBKE.
- 2802 7 The Trust Center shall use a device's install code key to transport the NWK key if a Trust
2803 Center rejoin is performed by a device from another profile.
2804

2805 **5.13.2 Requirements for Smart Energy Devices**

- 2806 1 Smart Energy endpoints that receive Smart Energy cluster messages using a profile other
2807 than the Smart Energy profile shall drop the messages.

- 2808 **2** Smart Energy endpoints must continue to follow the cluster security requirements specified
2809 in Table 5-13 and drop Smart Energy messages that do not have APS security when it is
2810 required (refer to section 5.4.6 for further details).
- 2811 **3** Smart Energy endpoints that also implement clusters from another profile shall not use their
2812 install code key for any encryption or decryption once they establish a CBKE link key.
- 2813 **4** Smart Energy endpoints shall not refer to a Smart Energy-related cluster on a non-Smart-
2814 Energy endpoint. Note that this does not preclude a non-Smart-Energy version of a cluster
2815 existing on a HAN as well as a Smart Energy-related version.

2816

2817 **5.13.3 Requirements for Devices from other Profiles**

2818 **5.13.3.1 Joining**

- 2819 **1** Devices from other profiles must implement install code based link keys. This includes
2820 proper documentation of the install code via packaging and instructions provided to the
2821 installer. Refer to section 5.4.8 for further details.
- 2822 **2** The Smart Energy network must be informed of the device that is to be joined. This is done
2823 through an out of band means which could include a web login, phone call to a service
2824 center, or handheld tool. Using this methodology the existing network is made aware of the
2825 device ID and Install Code/Link Key for the device.
- 2826 **3** The Smart Energy network is put into permit joining ON for a period of time.
- 2827 **4** The installer/homeowner is prompted to press a button or complete a menu sequence that
2828 tells the device to attempt to join a network.
- 2829 **5** The device joins the network.
- 2830 **6** An indicator is provided for the installer/homeowner indicating the device has joined a
2831 network.
- 2832 **7** The device can now operate normally on the network according to the rules of the relevant
2833 profile.
- 2834 **8** Devices from other profiles shall have no expectations they may use their link key with the
2835 Trust Center other than to encrypt/decrypt APS command frames. Their link key will be
2836 treated like an unauthorized link key. The Trust Center shall not accept APS data frames, for
2837 Smart Energy clusters requiring APS encryption, where such data frames are encrypted
2838 using an unauthorized link key.

2839 **9** Currently there is no provision in this specification for a device from another profile to
2840 obtain an authorized link key.

2841

2842 5.13.3.2 [Rejoining](#)

2843 Devices from other profiles may perform a secure or trust center rejoin to get back on the
2844 network. A trust center rejoin shall use the device's install code key.

2845

2846

6 Device Specifications

2847

6.1 Common Clusters

2848 Support for certain clusters is required on all products supporting this profile. At least one
 2849 instance of the clusters shown in Table 6-1 shall exist on a product supporting this profile.
 2850 Individual device descriptions may place further restrictions on support of the optional clusters
 2851 shown here. ZCL clusters not listed may be implemented on the Smart Energy endpoint.
 2852 Manufacturers may extend the SE profile as specified in the ZCL specification [B1].

2853

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

2854

2855

2856

6.1.1 Optional Support for Clusters with Reporting Capability

2857 Some clusters support the ability to report changes to the value of particular attributes. These
 2858 reports are typically received by the client side of the cluster. All devices in this profile may
 2859 support any cluster that receives attribute reports.

2860

6.1.2 Manufacturer-Specific Clusters

2861 The ZCL provides a range of cluster IDs that are reserved for manufacturer-specific
 2862 clusters. Manufacturer-specific clusters that conform to the requirements given in the ZCL may
 2863 be added to any device description specified in this profile.

2864

6.1.3 Cluster Usage Restrictions

2865 None.

2866 **6.1.4 Identify Cluster Best Practices**

2867 To help aid in locating devices, it's strongly recommended that all devices utilize the Identify
 2868 Cluster and a visual or audible indicator. In situations in which a device can't supply a visual
 2869 or audible indicator, the device should include a visible label with the appropriate information
 2870 to help identify the device.

2871 **6.1.5 Inter-PAN Communication**

2872 Inter-PAN access to Smart Energy devices shall be limited to specific clusters and commands.
 2873 Please refer to Annex B for further details.

2874 **6.2 Feature and Function Description**

2875 Each device must support a certain set of features and functions. Table 6-2 below is used to
 2876 specify the mandatory and optional features and functions for Smart Energy devices. This
 2877 chapter contains a description of what must be supported if the feature or function is supported
 2878 by the device. The mandatory or optional configuration for each device is described in the
 2879 upcoming chapters:

2880 **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

2881
 2882
 2883 **Join (End Devices and Routers):**
 2884 As described in sub-clauses 5.4 and 5.5.

2885 **Form Network (Coordinator):**

2886 As described in sub-clauses 5.4 and 5.5.

2887 **Allow Others to Join Network (Router and Coordinator Only):**

2888 As described in sub-clauses 5.4 and 5.5.

2889 **Restore to Factory Fresh Settings:**

2890 The Device shall provide a way to restore Factory Settings.

2891 **Pair Devices (End Device Bind Request):**

2892 Whenever possible, the device should provide a way for the user to issue an End Device Bind
2893 Request.

2894 **Bind Manager (End Device Bind Response – Coordinator only):**

2895 The coordinator device shall be capable of issuing an End Device Bind Response.

2896 **Enable Identify Mode:**

2897 Whenever possible, the device should provide a way for the user to enable Identify for 60
2898 seconds.

2899 **Service Discovery (Match Descriptor Request):**

2900 Whenever possible, the device should provide a way for device to send a match descriptor
2901 request, receive match descriptor responses and utilize them for commissioning the device.

2902 **ZDP Bind Response:**

2903 The device shall be able to receive a ZDP Bind Request and respond correctly with a ZDP
2904 Bind Response.

2905 **ZDP Unbind Response:**

2906 The device shall be able to receive a ZDP Unbind Request and respond correctly with a ZDP
2907 Unbind Response.

2908 **End Device Annce/Device Annce:**

2909 The device shall Send End Device Annce / Send Device upon joining and re- joining a
2910 network.

2911 **Service Discovery Response:**

2912 The Device shall be able to receive a Match descriptor request, and respond with a match
2913 descriptor response correctly.

2914 **Allow Smart Energy Devices to Join the Network:**

2915 The Device shall allow other Smart Energy devices to join the network.

2916 **High Security Supported:** No

2917 **Enhanced Inter-PAN Communication:**

2918 The device may support Enhanced Inter-PAN Communications as described in Annex B

2919 **6.3 Smart Energy Devices**

2920 A physical device may support one or more logical Smart Energy devices. The supported
 2921 clusters of a logical Smart Energy device shall reside on a single endpoint, with the
 2922 exception of the common clusters listed in Table 6-1, which may reside on a separate endpoint
 2923 using the Physical Device identifier. Each logical Smart Energy device on a single physical
 2924 device shall reside on its own separate endpoint.

2925 SE devices shall use the device and service discovery mechanisms specified in the ZigBee
 2926 specification [B3] to find the services required. Devices shall support discovery of single and
 2927 multiple endpoints on a single physical device. In the case where multiple devices of the same
 2928 type are discovered, SE cluster attributes should be read to determine the type of service
 2929 provided. For example, if multiple Metering devices are discovered the *MeteringDeviceType*
 2930 attribute provides a label for identifying the type of metering device present. Similarly, if
 2931 multiple ESIs are found the *CommodityType* attribute shall be read to determine the fuel type
 2932 of that ESI.

2933 **6.3.1 Energy Service Interface**

2934 The Energy Service Interface connects the energy supply company communication
 2935 network to the metering and energy management devices within the home. It routes messages
 2936 to and from the relevant end points. It may be installed within a meter, thermostat, or In-
 2937 Home Display, or may be a standalone device, and it will contain another non-ZigBee
 2938 communication module (e.g. power-line carrier, RF, GPRS, broadband Internet connection).

2939 **6.3.1.1 Supported Clusters**

2940 In addition to those specified in Table 6-1, the Energy Service Interface device shall support the
 2941 clusters listed in Table 6-3. If a SE cluster is not listed as mandatory or optional in the
 2942 following table or in the common table, then that cluster shall be prohibited on an ESI device
 2943 endpoint.

2944 **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

2945

2946

2947 **6.3.1.2 Supported Features and Functions**

2948 The Energy Service Interface device shall have the features and functions listed in Table 6-2.

2949 **6.3.2 Metering Device**2950 The Metering end device is a meter (electricity, gas, water, heat, etc.) that is fitted with a ZigBee
2951 device. Depending on what is being metered, the device may be capable of immediate
2952 (requested) reads or it will autonomously send readings periodically. A Metering end device
2953 may also be capable of communicating certain status indicators (e.g. battery low, tamper
2954 detected).2955 **6.3.2.1 Supported Clusters**2956 In addition to those specified in Table 6-1, the Metering Device shall support the clusters listed
2957 in Table 6-4. If a SE cluster is not listed as mandatory or optional in the following table or in the
2958 common table, then that cluster shall be prohibited on a Metering device endpoint.2959 **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

2960

2961 [6.3.2.2 Supported Features and Functions](#)

2962 The Metering Device shall have the features and functions listed in Table 6-2.

2963

2964 **6.3.3 In-Home Display Device**

2965 The In-Home Display device will relay energy consumption data to the user by way of a
 2966 graphical or text display. The display may or may not be an interactive device. At a minimum at
 2967 least one of the following should be displayed: current energy usage, a history over
 2968 selectable periods, pricing information, or text messages. As an interactive device, it can be
 2969 used for returning simple messages for interpretation by the recipient (e.g. "Button A was
 2970 pressed").

2971 The display may also show critical pricing information to advise the customer when peaks are
 2972 due to occur so that they can take appropriate action.

2973 **6.3.3.1 Supported Clusters**

2974 In addition to those specified in Table 6-1, the In-Home Display device shall support the clusters
 2975 listed in Table 6-5. If a SE cluster is not listed as mandatory or optional in the following table or
 2976 in the common table, then that cluster shall be prohibited on an In-Home Display device endpoint.

2977 **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

2978
 2979 An In-Home Display shall implement at least one of the optional client clusters listed.

2980 **6.3.3.2 Supported Features and Functions**

2981 The In-Home Display device shall have the features and functions listed in Table 6-2.

2982

2983 **6.3.4 Programmable Communicating Thermostat (PCT) Device**2984 The PCT device shall provide the capability to control the premises heating and cooling
2985 systems.2986 **6.3.4.1 Supported Clusters**2987 In addition to those specified in Table 6-1, the PCT device shall support the clusters listed in
2988 Table 6-6. If a SE cluster is not listed as mandatory or optional in the following table or in the
2989 common table, then that cluster shall be prohibited on a PCT device endpoint.2990 **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

2991

2992 **6.3.4.2 Supported Features and Functions**

2993 The PCT device shall have the features and functions listed in Table 6-2.

2994 **6.3.5 Load Control Device**2995 The Load Control device is capable of receiving Demand Response and Load Control events
2996 to manage consumption on a range of devices. Example devices are water heaters, exterior
2997 lighting, and pool pumps.

2998 **6.3.5.1 Supported Clusters**

2999 In addition to those specified in Table 6-1, the Load Control device shall support the clusters
 3000 listed in Table 6-7.

3001 **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

3002

3003 **6.3.5.2 Supported Features and Functions**

3004 The Load Control Device shall support the features and functions listed in Table 6-2.

3005 **6.3.6 Range Extender Device**

3006 The Range Extender is a simple device that acts as a router for other devices. The Range
 3007 Extender device shall not be a ZigBee end device. A product that implements the Range
 3008 Extender device shall not implement any other devices defined in this profile. This device shall
 3009 only be used if the product is not intended to have any other application, or if a private
 3010 application is implemented that has not been addressed by this profile.

3011 **6.3.6.1 Supported Clusters**

3012 The Range Extender device shall support the mandatory common clusters listed in Table 6-1.

3013 **6.3.6.2 Supported Features and Functions**

3014 The Range Extender device shall have the features and functions listed in Table 6-2.

3015 **6.3.7 Smart Appliance Device**

3016 Smart Appliance devices on the ZigBee network can participate in energy management
 3017 activities. Examples of these are when Utilities initiate a demand response or pricing event, or

3018 the appliance actively informs customers via in-home displays of when or how energy is
 3019 being used. In the latter case, scenarios include:

- 3020 • Washer switching to cold water during periods of higher energy costs.
 3021 • Washer/Dryer/Oven/Hot Water Heater reporting cycle status.
 3022 • Over temperature conditions in Freezers and Refrigerators.

3023 **6.3.7.1 Supported Clusters**

3024 In addition to those specified in Table 6-1 the Smart Appliance device shall support the
 3025 clusters listed in Table 6-8. If a SE cluster is not listed as mandatory or optional in the following
 3026 table or in the common table, then that cluster shall be prohibited on a Smart Appliance device
 3027 endpoint.

3028 **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

3029

3030

3031 **6.3.7.2 Supported Features and Functions**

3032 The Smart Appliance device shall have the features and functions listed in Table 6-2.

3033

3034 **6.3.8 Prepayment Terminal Device**

3035 The Prepayment Terminal device will allow utility customers or other users (e.g. sub-metered
 3036 tenants) to pay for consumption in discrete increments rather than establishing a traditional

3037 billing agreement. The Prepayment Terminal device will accept payment (e.g. credit card, code
 3038 entry), display remaining balances, and alert the user of a balance approaching zero, and may
 3039 perform some or all of the other functions described in sub-clause 6.3.3 “In-Home Display
 3040 Device”.

3041 **6.3.8.1 Supported Clusters**

3042 In addition to those specified in Table 6-1, the Prepayment Terminal device shall support the
 3043 clusters listed in Table 6-9. If a SE cluster is not listed as mandatory or optional in the following
 3044 table or in the common table, then that cluster shall be prohibited on a Prepayment Terminal
 3045 device endpoint.

3046 **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

3047 **6.3.8.2 Supported Features and Functions**

3048 The Prepayment Terminal device shall have the features and functions listed in Table 6-2.

3049 **6.3.9 Physical Device**

3050 The Physical Device type will identify a supplemental (or sole) endpoint on which the clusters
 3051 related to a physical product may reside. The endpoint shall not contain any cluster related
 3052 to any individual logical SE device on the physical product. A product is allowed to have a
 3053 Physical Device as its sole SE endpoint. A Physical Device must be capable of providing other
 3054 SE device endpoints to be a certified SE product.
 3055

3056 [6.3.9.1 Supported Clusters](#)

3057 The Physical Device may only support the common clusters listed in Table 6-1.

3058 [6.3.9.2 Supported Features and Functions](#)

3059 The Physical Device shall have the features and functions listed in Table 6-2.

3060

3061 [6.3.10 Remote Communications Device](#)

3062 The Remote Communications Device may be used to provide a WAN communication path
 3063 through the ZigBee HAN where an ESI does not have a direct WAN connection. The Remote
 3064 Communications Device endpoint may reside on a physical device with a direct connection to the
 3065 WAN, allowing it to tunnel WAN protocol messages to an ESI which does not have a direct
 3066 WAN connection.

3067 [6.3.10.1 Supported Clusters](#)

3068 In addition to those specified in Table 6-1, the Remote Communications Device shall support
 3069 the mandatory clusters listed in Table 6-10 and may support the optional clusters. If a SE cluster
 3070 is not listed as mandatory or optional in the following table or in the common table, then that
 3071 cluster shall be prohibited on a Remote Communications Device endpoint.

3072 **Table 6-10 – Clusters Supported by the Remote Communications Device**

Server Side	Client Side
Mandatory	
Optional	
Tunneling	Tunneling
Time	

3073

3074 A Remote Communications Device shall implement at least one of either the Tunneling cluster
 3075 Server or Client.

3076 Provision of the optional Time server allows support of an accurate commodity-agnostic time
 3077 source, if such a source were available via the WAN connection.

3078 [6.3.10.2 Supported Features and Functions](#)

3079 The Remote Communications device shall have the features and functions listed in Table 6-2.

3080

3081

3082

3083 **Annex A CANDIDATE ZCL MATERIAL FOR USE WITH THIS PROFILE**

3084 The candidate material in this annex, when approved, will be merged into the Foundation
 3085 document of the ZigBee Cluster Library (ZCL) by the Cluster Library Development Board.

3086 **A.1 New Status Enumeration**

3087 A new ZCL status enumeration is required for use with Smart Energy devices, as described in
 3088 Table A-1.

3089 **Table A-1– Additional Status Enumeration**

3089 Enumerated Status	Value	Description
NOTIFICATION_PENDING	0x9A	The command has been received and is being processed.

3090

3091 **A.2 New Attribute Reporting Status Indication**

3092 A new *Attribute Reporting Status* attribute is to be added to all Smart Energy clusters.
 3093 Consideration will be given to extending this to all ZCL clusters. Details for this attribute are
 3094 shown in Table A-2:

3095 **Table A-2– Attribute Reporting Status Attribute**

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

3096

3097 **A.2.1 Attribute Reporting Status Attribute**

3098 When utilizing the *Report Attributes* command, this 8-bit enumeration should be included as the last
 3099 attribute in order to indicate that all required attributes have been reported. The enumerated values
 3100 for this attribute are outlined in Table A-3:

3101 **Table A-3– Attribute Reporting Status Enumerations**

3101 Enumerated Value	Status
0x00	Pending
0x01	Attribute Reporting Complete
0x02 to 0xFF	Reserved for future use

3102

3103

3104

3105

3106

3107

Annex B ENHANCED INTER-PAN TRANSMISSION MECHANISM

3108 NOTE: This annex previously described the requirements for the 'Anonymous Inter-PAN'
3109 mechanism. It has been removed and replaced with a new annex defining the 'Enhanced' Inter-
3110 PAN mechanism which is to be used in conjunction with devices such as Hand Held Terminals
3111 (HHTs).

3112

B.1 Scope and Purpose

3113 This annex defines a mechanism whereby ZigBee devices can perform limited exchanges of
3114 information with devices in their local neighborhood without having to form or join the same
3115 ZigBee network. The Enhanced Inter-PAN mechanism is intended to be used in conjunction with
3116 devices such as Hand Held Terminals (HHTs).

3117 The intended destination for the mechanism described here is not the ZigBee specification
3118 [B7], but the relevant application profile documents for applications that make use of the
3119 feature – in particular, the Smart Energy Standard.

3120

B.2 General Description

B.2.1 What Enhanced Inter-PAN Transmission Does

3121 A schematic view of the how inter-PAN transmission in a ZigBee context works is shown in
3122 Figure B-1.

3124

3125

3126

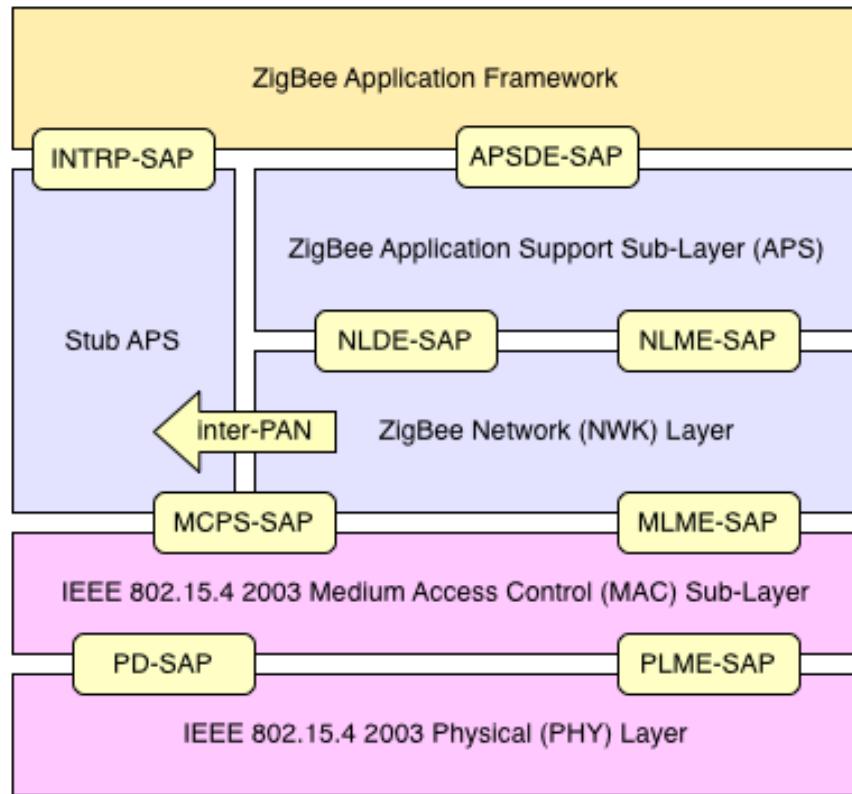


Figure B-1– ZigBee Stack with Stub APS

3127
3128
3129

3130 Inter-PAN data exchanges are handled by a special “stub” of the Application Support Sub-
 3131 Layer, which is accessible through a special Service Access Point (SAP), the INTRP-SAP, parallel to the normal APSDE-SAP. The stub APS performs just enough processing to pass
 3132 application data frames to the MAC for transmission and also to pass Inter-PAN application
 3133 frames from the MAC to the application on receipt.

3135 The Inter-PAN data exchange architecture does not support simultaneous execution by multiple
 3136 application entities. Within a device, only one application entity shall use the Inter-PAN
 3137 communications mechanisms.

3138 The Enhanced Inter-PAN specification includes options for its operation and security. Each
 3139 profile and feature that uses Inter-PAN will specify the options allowed.

3140 **B.3 Service Specification**

3141 The INTRP-SAP is a data service comprising three primitives.

- 3142 • INTRP-DATA.request - Provides a mechanism for a sending device to request transmission
3143 of an Inter-PAN message.
- 3144 • INTRP-DATA.confirm - Provides a mechanism for a sending device to understand the status
3145 of a previous request to send an Inter-PAN message.
- 3146 • INTRP-DATA.indication - Provides a mechanism for identifying and conveying an Inter-
3147 PAN message received from a sending device.

3148 **B.3.1 The INTRP-DATA.request Primitive**

3149 The INTRP-DATA.request primitive allows an application entity to request data transmission
3150 via the stub APS.

3151 **B.3.1.1 Semantics of the Service Primitive**

3152 The primitive interface is as follows:

3153

3154 INTRP-DATA.request	{
3155 SrcAddrMode	
3156 DstAddrMode	
3157 DstPANId	
3158 DstAddress	
3159 ProfileId	
3160 ClusterId	
3161 ASDULength	
3162 ASDU	
3163 ASDUHandle	
3164 TxOptions	
3165 }	

3166

3167 Parameters of the primitive appear in Table B-1.

3168

3169

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)

3170
3171

3172 **B.3.1.2 When Generated**

3173 This primitive is generated by the local application entity when it wishes to address a frame
 3174 to one or more peer application entities residing on neighboring devices with which it does not
 3175 share a network association.

3176

3177 **B.3.1.3 Effect on Receipt**

3178 On receipt of the INTRP-DATA.request primitive by the stub APS, the stub APS will construct
 3179 and transmit a frame containing the given ASDU and other parameters using the MCPS-
 3180 DATA.request primitive of the MAC sub-layer, as described in sub-clause B.5.1, and, once the
 3181 corresponding MCPS-DATA.confirm primitive is received, generate the INTRP-DATA.confirm
 3182 primitive with a status value reflecting the status value returned by the MAC.

3183 **B.3.2 The INTRP-DATA.confirm Primitive**

3184 The INTRP-DATA.confirm primitive allows the stub APS to inform the application entity about
 3185 the status of a data request.

3186 **B.3.2.1 Semantics of the Service Primitive**

3187 The primitive interface is as follows:

INTRP-DATA.confirm { ASDUHandle Status }

3192 Parameters of the primitive appear in Table B-2.

3193 **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.

3194

3195 **B.3.2.2 When Generated**

3196 This primitive is generated by the stub APS on a ZigBee device and passed to the application in
3197 response to the receipt of a MCPS-DATA.confirm primitive that is a confirmation of a previous
3198 MCPS-DATA.request issued by the stub APS.

3199 **B.3.2.3 Effect on Receipt**

3200 As a result of the receipt of this primitive, the application is informed of the results of an
3201 attempt to send a frame via the stub APS.

3202 **B.3.3 The INTRP-DATA.indication Primitive**

3203 The INTRP-DATA.indication primitive allows the stub APS to inform the next higher layer
3204 that it has received a frame that was transmitted via the stub APS on another device.

3205 **B.3.3.1 Semantics of the Service Primitive**

3206 The primitive interface is as follows:

```
3207
3208    INTRP-DATA.indication {  
3209         SrcAddrMode  
3210         SrcPANId  
3211         SrcAddress  
3212         DstAddrMode  
3213         DstPANId  
3214         DstAddress  
3215         ProfileId  
3216         ClusterId  
3217         ASDULength  
3218         ASDU  
3219         Status  
3220         SecurityStatus  
3221         LinkQuality  
3222     }  
3223
```

3224

3225 Parameters of the primitive appear in Table B-3.

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

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

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.

3227

3228 **B.3.3.2 When Generated**

3229 This primitive is generated and passed to the application in the event of the receipt, by the
 3230 stub APS, of a MCPS-DATA.indication primitive from the MAC sub-layer, containing a frame
 3231 that was generated by the stub APS of a peer ZigBee device, and that was intended for the
 3232 receiving device.

3233 **B.3.3.3 Effect on Receipt**

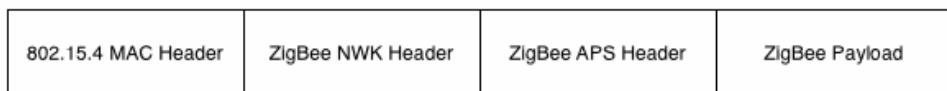
3234 Upon receipt of this primitive the application is informed of the receipt of an application
 3235 frame transmitted, via the stub APS, by a peer device and intended for the receiving device.

3236 **B.3.4 Qualifying and Testing of Inter-PAN Messages**

3237 Certification and application level testing shall ensure both the sending and receiving devices
 3238 correctly react and understand the INTRP-DATA.request and INTRP-DATA.indication
 3239 primitives.

3240 **B.4 Frame Formats**

3241 The birds-eye view of a normal ZigBee frame is as shown in Figure B-2.

3242
3243 **Figure B-2– Normal ZigBee Frame**

3244 Briefly, the frame contains the familiar headers controlling the operation of the MAC sub-
 3245 layer, the NWK layer and the APS. Following these, there is a payload, formatted as specified in
 3246 [B1].

3247 Since most of the information contained in the NWK and APS headers is not relevant for
 3248 Inter-PAN transmission, the Inter-PAN frame, shown Figure B-3, contains only a stub of the
 3249 NWK header and the APS header, which provide the information required by the stub APS
 3250 shown in Figure B-4 to do its job.

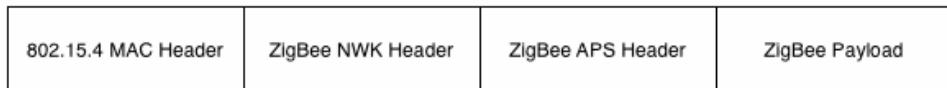


Figure B-3– Enhanced Inter-PAN ZigBee Frame

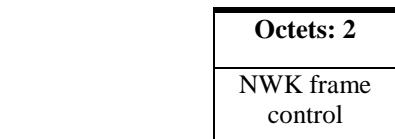


Figure B-4– Stub NWK Header Format

3256 The format of the frame control field of the stub NWK header is formatted as shown in Figure
 3257 B-5.

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

Figure B-5– NWK Frame Control Field

3259 The sub-fields of the NWK frame control field are as follows:

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

3264 All other sub-fields shall have a value of 0.

3265 The format of the stub APS header is shown in Figure B-6.

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

Figure B-6– Stub APS Header Format

3267 The stub APS header contains as many as seven fields totaling a maximum of 25 octets in
 3268 length.

3269 The APS frame control field shall be 1 octet in length and is identical in format to the frame
 3270 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

3271 **Figure B-7– APS Frame Control Field**

3272 The fields of the frame control field have the following values:

- 3273 • The frame type sub-field shall have a value of 0b11, which is a reserved frame type with
 3274 respect to the [B3].
- 3275 • The delivery mode sub-field may have a value of 0b00, indicating unicast, 0b10, indicating
 3276 broadcast or 0b11 indicating group addressing.
- 3277 • The Security sub-field may have the value 0 for no security or 1 for a secured frame. When
 3278 the Security sub-field is 1 an Auxiliary Header shall be present.
- 3279 • The ACK request sub-field shall have a value of 0 when no ACK is requested and value 1
 3280 when an ACK is requested. When an ACK is requested, the APS Counter field shall be present.
- 3281 • The extended header present sub-field shall have a value of 0, indicating no extended header,
 3282 or a value of 1, indicating that a message has been fragmented and that there is an extended
 3283 header.

3284 The optional Group Address shall be present if and only if the Delivery Mode field has a value
 3285 of 0x0b11 and the delivery mode is 0b11 indicating a group address. If present, it shall contain
 3286 the 16-bit identifier of the group to which the frame is addressed.

3287 The Cluster Identifier field is 2 octets in length and specifies the identifier of the cluster to
 3288 which the frame relates and which shall be made available for filtering and interpretation of
 3289 messages at each device that takes delivery of the frame.

3290 The Profile Identifier is two octets in length and specifies the ZigBee profile identifier for
 3291 which the frame is intended and shall be used during the filtering of messages at each device that
 3292 takes delivery of the frame. When the Profile Identifier is set to that of the Smart Energy profile,
 3293 the Security sub-field of the APS Control field shall be set as follows. If the cluster ID is set to
 3294 0x0019 (Key Establishment), the Security sub-field shall be set to 0; otherwise it shall be set to 1.

3295 **B.5 Frame Processing**

3296 Assuming the INTRP-SAP described above, frames transmitted using the stub APS are
3297 processed as described here.

3298 **B.5.1 Enhanced Inter-PAN Transmission**

3299 On receipt of the INTRP-DATA.request primitive, the stub APS shall construct a stub APS
3300 frame. The header of the stub APS frame shall contain a NWK and an APS frame control field
3301 as described in clause B.4, a cluster identifier field equal to the value of the ClusterId parameter
3302 of the INTRP-DATA.request and a profile identifier field equal to the value of the ProfileId
3303 parameter. If the DstAddrMode parameter of the INTRP-DATA.request has a value of 0x01,
3304 indicating group addressing, then the APS header shall also contain a group address field with
3305 a value corresponding to the value of the DstAddress parameter. The payload of the stub APS
3306 frame shall contain the data payload to be transmitted.

3307 The stub APS frame will then be transmitted using the MCPS-DATA.request primitive of the
3308 MAC sub-layer with key primitive parameters set as follows:

- 3309 • The value of the SrcAddrMode parameter of the MCPS-DATA.request shall always be set to
3310 a value of three, indicating the use of the 64-bit extended address.
- 3311 • The SrcPANId parameter shall be equal to the value of the *macPANID* attribute of the MAC
3312 PIB.
- 3313 • The SrcAddr parameter shall always be equal to the value of the MAC sub- layer constant
3314 *aExtendedAddress*.
- 3315 • If the DstAddrMode parameter of the INTRP-DATA.request primitive has a value of 0x01,
3316 then the DstADdrMode parameter of the MCPS-DATA.request shall have a value of 0x02.
3317 Otherwise, the DstAddrMode parameter of the MCPS-DATA.request shall reflect the value
3318 of the DstAddrMode parameter of the INTRP-DATA.request.
- 3319 • The DstPANId parameter shall have the value given by the DstPANID parameter of the
3320 INTRP-DATA.request primitive.
- 3321 • If the DstAddrMode parameter of the INTRP-DATA.request has a value of 0x01, indicating
3322 group addressing, then the value of the DstAddr parameter of the MCPS-DATA.request shall
3323 be the broadcast address 0xffff. Otherwise, the value of the DstAddr parameter shall reflect
3324 the value of the DstAddress parameter of the INTRP-DATA.request primitive.
- 3325 • The MsduLength parameter shall be the length, in octets, of the stub APS frame.
- 3326 • The Msdu parameter shall be the stub APS frame itself.

- 3327 • If the transmission is a unicast packet, then the value of the TxOptions parameter shall be
3328 0x01, indicating a request for acknowledgement. Otherwise, the TxOptions parameter shall
3329 have a value of 0x00, indicating no options.

3330 On receipt of the MCPS-DATA.confirm primitive from the MAC sub-layer, the stub APS will
3331 invoke the transmit confirmation function with a status reflecting the status returned by the
3332 MAC.

3333 **B.5.2 Enhanced Inter-PAN Reception**

3334 On receipt of the MCPS-DATA.indication primitive from the MAC sub-layer, the receiving
3335 entity - in the case of a ZigBee device this is normally the NWK layer - shall determine
3336 whether the frame should be passed to the stub APS or processed as specified in [B3]. For a
3337 frame that is to be processed by the stub APS, the non- varying sub-fields of both the NWK
3338 frame control field and the APS frame control field must be set exactly as described above.

3339 If the delivery mode sub-field of the APS frame control field of the stub APS header has a
3340 value of 0b11, indicating group addressing, then, if the device implements group
3341 addressing, the value of the group address field shall be checked against the NWK layer
3342 group table, and, if the received value is not present in the table, the frame shall be discarded
3343 with no further processing or action.

3344 On receipt of a frame for processing, the stub APS shall generate an INTRP- DATA.indication
3345 with parameter values as follows:

- 3346 • The value of the SrcAddrMode parameter of the INTRP-DATA.indication shall always be
3347 set to a value of three, indicating the use of the 64-bit extended address
- 3348 • The value of the SrcPANId parameter shall reflect that of the SrcPANId parameter of the
3349 MCPS-DATA.indication.
- 3350 • The SrcAddress parameter of the INTRP-DATA.indication shall always reflect the value of a
3351 64-bit extended address.
- 3352 • Values for the DstAddrMode parameter shall be one of the following:
 - 3353 ○ 0x03, if the DstAddrMode parameter of the INTRP-DATA.indication has a value of
3354 0x03.
 - 3355 ○ 0x02, if the DstAddrMode parameter of the INTRP-DATA.indication has a value of
3356 0x02
- 3357 • The value of the DstPANId parameter of the INTRP-DATA.indication shall reflect the value
3358 of the DstPANId parameter of the MCPS-DATA.indication.

- 3359 • If the DstAddrMode parameter of the INTRP-DATA.indication has a value of 0x01,
3360 indicating group addressing, then the DstAddress parameter of the INTRP-DATA.indication
3361 shall reflect the value of the Group Address field of the stub APS header. Otherwise, the
3362 value of the DstAddress parameter of the INTRP-DATA.indication shall reflect the value of
3363 the DstAddr parameter of the MCPS-DATA.indication.
- 3364 • The value of the ProfileId parameter shall be the same as the value of the Profile Identifier
3365 field of the stub APS header.
- 3366 • The value of the ClusterId parameter shall be the same as the value of the Cluster Identifier
3367 field of the stub APS header.
- 3368 • The ASDULength field shall contain the number of octets in the stub APS frame payload.
- 3369 • The ASDU shall be the stub APS payload itself.
- 3370 • The value of the LinkQuality parameter shall reflect the value of the mpduLinkQuality
3371 parameter of the MCPS-DATA.indication.

3372 **B.6 Initiating a Enhanced Inter-PAN Interaction**

- 3373 1. Unless Inter-PAN is being used in conjunction with Network Joining, it is assumed that there
3374 will be no need to permit joining on the network on which the required receiving device is
3375 present.
- 3376 2. To find the required receiving device with which to interact, the initiating device shall send
3377 Beacon Request packets on each channel, dwelling on each channel as specified by the
3378 ZigBee PRO specification beacon response window.
- 3379 3. The device shall survey all required channels and build a list of devices which have emitted
3380 a beacon that has been heard. If the initiating device has a user interface, the available
3381 devices should be displayed in order that the required receiving device can be chosen. If no
3382 user interface is available, then other method(s) shall be employed to choose a required
3383 receiving device (e.g. strongest signal).
- 3384 4. The device initiates a CBKE process with the required receiving device by sending an
3385 *Initiate Key Establishment Request* command within an unsecured Enhanced Inter-PAN
3386 frame (Security sub-field set to 0).
- 3387 5. The required receiving device may employ techniques to ensure that the initiating device is
3388 acceptable for such communication (e.g. by checking its EUI-64 address against a list), and
3389 refusing the transaction if it is not suitable.

- 3390 6. If the initiating device is acceptable to the required receiving device, the Enhanced Inter-
3391 PAN CBKE process will be allowed to complete, and the resultant shared APS link key shall
3392 be used to secure all further Enhanced Inter-PAN frames (Security sub-field set to 1).

3393 **B.7 Best Practices**

3394 Network Channel Manager Inter-PAN support is not specified in Annex E of the core stack
3395 specification ([B3]). New channel notifications will not be broadcast using Inter-PAN. Inter-
3396 PAN devices which do not receive the network channel change will need to perform the network
3397 discovery procedure described in B.3.4.

3398 **B.8 Security Requirements**

3399 Smart Energy profile devices supporting Enhanced Inter-PAN shall use APS security for all Inter-
3400 PAN frames except for those key establishment messages that do not require it.

3401 In addition, devices shall verify the correct format of all SE messages received over Enhanced
3402 Inter-PAN. Any received message that does not conform to the format described in this
3403 document shall be dropped.

3404

3405

Annex C KEY ESTABLISHMENT CLUSTER

3406

3407
3408

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

3409

C.1 Scope and Purpose

3410

3411

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

3412

3413
3414

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

3415

3416
3417

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

3418

C.2 General Description

3419

C.2.1 Introduction

3420

3421

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

3422

C.2.2 Security Credentials

3423

3424
3425
3426
3427

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

3428

C.2.3 Network Security

3429

3430
3431

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

3432

C.2.4 Key Establishment

3433

3434
3435

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

3436 AES-CCM*. This relies on the existence of secret keying material shared between the involved
3437 devices. There are methods to distribute this secret keying material in a trusted manner. However,
3438 these methods are generally not scalable or communication may be required with a trusted key
3439 allocation party over an insecure medium. This leads to the requirement for automated key
3440 establishment schemes to overcome these problems.

3441 Key establishment schemes can either be effected using either a key agreement scheme or a
3442 key transport scheme. The key establishment scheme described in this document uses a key
3443 agreement scheme, therefore key transport schemes will not be considered further in this
3444 document.

3445 A key agreement scheme is where both parties contribute to the shared secret and therefore the
3446 secret keying material to be established is not sent directly; rather, information is exchanged
3447 between both parties that allows each party to derive the secret keying material. Key agreement
3448 schemes may use either symmetric key or asymmetric key (public key) techniques. The party
3449 that begins a key agreement scheme is called the initiator, and the other party is called the
3450 responder.

3451 Key establishment using key agreement involves an initiator and a responder and four steps:

- 3452 1 Establishment of a trust relationship
- 3453 2 Exchange of ephemeral data
- 3454 3 Use of this ephemeral data to derive secret keying material using key agreement
- 3455 4 Confirmation of the secret keying material.

3456 There are two basic types of key establishment which can be implemented:

- 3457 • Symmetric Key Key Establishment
- 3458 • Public Key Key Establishment

3459 **C.2.5 Symmetric Key Key Establishment**

3460 Symmetric Key Key Establishment (SKKE) is based upon establishing a link key based on a
3461 shared secret (master key). If the knowledge of the shared secret is compromised, the
3462 established link key can also be compromised. If the master key is publicly known or is set to a
3463 default value, it is known as Unprotected Key Establishment (UKE). SKKE is the key
3464 establishment method used in the ZigBee specification therefore it will not be considered any
3465 further.

3466 **C.2.6 Public Key Key Establishment**

3467 Public Key Key Establishment (PKKE) is based upon establishing a link key based on shared
3468 static and ephemeral public keys. As the public keys do not require any secrecy, the
3469 established link key cannot be compromised by knowledge of them.

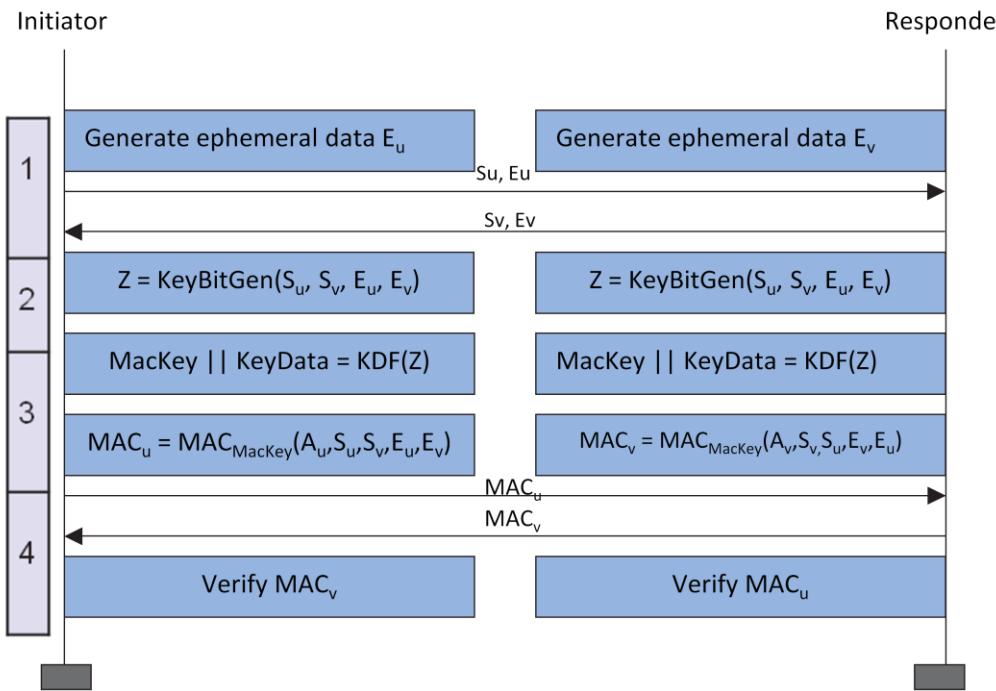
3470 As a device's static public key is used as part of the link key creation, it can either be transported
3471 independently to the device's identity where binding between the two is assumed, or it can be
3472 transported as part of a implicit certificate signed by a Certificate Authority, which provides
3473 authentication of the binding between the device's identity and its public key as part of the key
3474 establishment process. This is called Certificate-Based Key Establishment (CBKE) and is
3475 discussed in more detail in sub-clause C.4.2.

3476 CBKE provides the most comprehensive form of Key Establishment and therefore will be the
3477 method specified in this cluster.

3478 The purpose of the key agreement scheme as described in this document is to produce shared
3479 secret keying material which can be subsequently used by devices using AES-CCM* the
3480 cryptographic scheme deployed in the ZigBee Specification or for any proprietary
3481 security mechanism implemented by the application.

3482 **C.2.7 General Exchange**

3483 The following diagram shows an overview of the general exchange which takes place between
3484 initiator and responder to perform key establishment.



3485

3486

Figure C-1– Overview of General Exchange

3487 The functions are as follows:

- 3488 1 Exchange Static and Ephemeral Data
- 3489 2 Generate Key Bitstream
- 3490 3 Derive MAC key and Key Data
- 3491 4 Confirm Key using MAC

3492 The functions shown in the diagram (Figure C-1) depend on the Key Establishment mechanism.

3493 **C.2.7.1 Exchange Static and Ephemeral Data**

3494 Figure C-1 shows static data S_u and S_v . For PKKE schemes, this represents a combination of
 3495 the 64-bit device address [B11] and the device's static public key. The identities are needed by
 3496 the MAC scheme and the static public keys are needed by the key agreement scheme.

3497 Figure C-1 also shows ephemeral data E_u and E_v . For PKKE schemes, this represents the
 3498 public key of a randomly generated key pair.

3499 The static and ephemeral data S_U and E_U are sent to V and the static and ephemeral data S_V and
3500 E_V and are sent to U .

3501 **C.2.7.2 Generate Key Bitstream**

3502 Figure C-1 shows the KeyBitGen function for generating the key bitstream. The function's four
3503 parameters are the identifiers and the ephemeral data for both devices. This ensures the same
3504 key is generated at both ends.

3505 For PKKE schemes, this is the ECMQV key agreement schemes specified in Section 6.2 of
3506 SEC1 [B18]. The static data S_U represents the static public key $Q_{1,U}$ of party U , the static data
3507 S_V represents the static public key $Q_{1,V}$ of party V , the ephemeral data E_U represents the
3508 ephemeral public key $Q_{2,U}$ of party U and the ephemeral data E_V represents the ephemeral
3509 public key $Q_{2,V}$ of party V .

3510 **C.2.7.3 Derive MAC Key and Key Data**

3511 Figure C-1 shows the KDF (KeyDerivation Function) for generating the MAC Key and key
3512 data. The MAC Key is used with a keyed hash message authentication function to generate a
3513 MAC and the key data is the shared secret, e.g. the link key itself required for frame protection.

3514 For PKKE schemes, this is the key derivation function as specified in Section 3.6.1 of SEC1
3515 [B18]. Note there is no *SharedInfo* parameter of the referenced KDF, i.e. it is a null octet string
3516 of length 0.

3517 Figure C-1 also shows generation of the MAC using the MAC Key derived using the KDF using
3518 a message comprised of both static data S_U and S_V and ephemeral data E_U and E_V plus an
3519 additional component A which is different for initiator and responder.

3521 For PKKE schemes, this is the MAC scheme specified in section 3.7 of SEC1 [B18]. The
3522 MAC in the reference is the keyed hash function for message authentication specified
3523 in sub-clause C.4.2.2.6 and the message M is a concatenation of the identity (the 64-bit
3524 device address [B11]) of U , the identity of V and point-compressed octet-string representations of
3525 the ephemeral public keys of parties U and V . The order of concatenation depends on whether
3526 it is the initiator or responder. The additional component A is the single octet 02_{16} for the
3527 initiator and 03_{16} for the responder.

3528 **C.2.7.4 Confirm Key Using MAC**

3529 Figure C-1 shows MACs MAC_U and MAC_V

3530 The MAC MAC_U is sent to V and the MAC MAC_V is sent to U . U and V both calculate the
3531 corresponding MAC and compare it with the data received.

3532 **C.3 Cluster List**

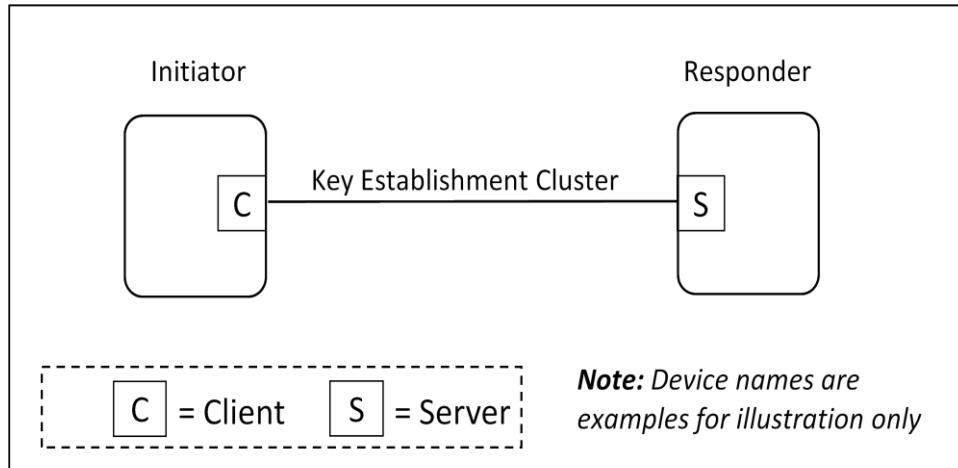
3533 The clusters specified in this document are listed in Table C-1.

3534 For our purposes, any device that implements the client side of this cluster may be considered the
3535 initiator of the secure communication transaction.

3536 **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.

3537



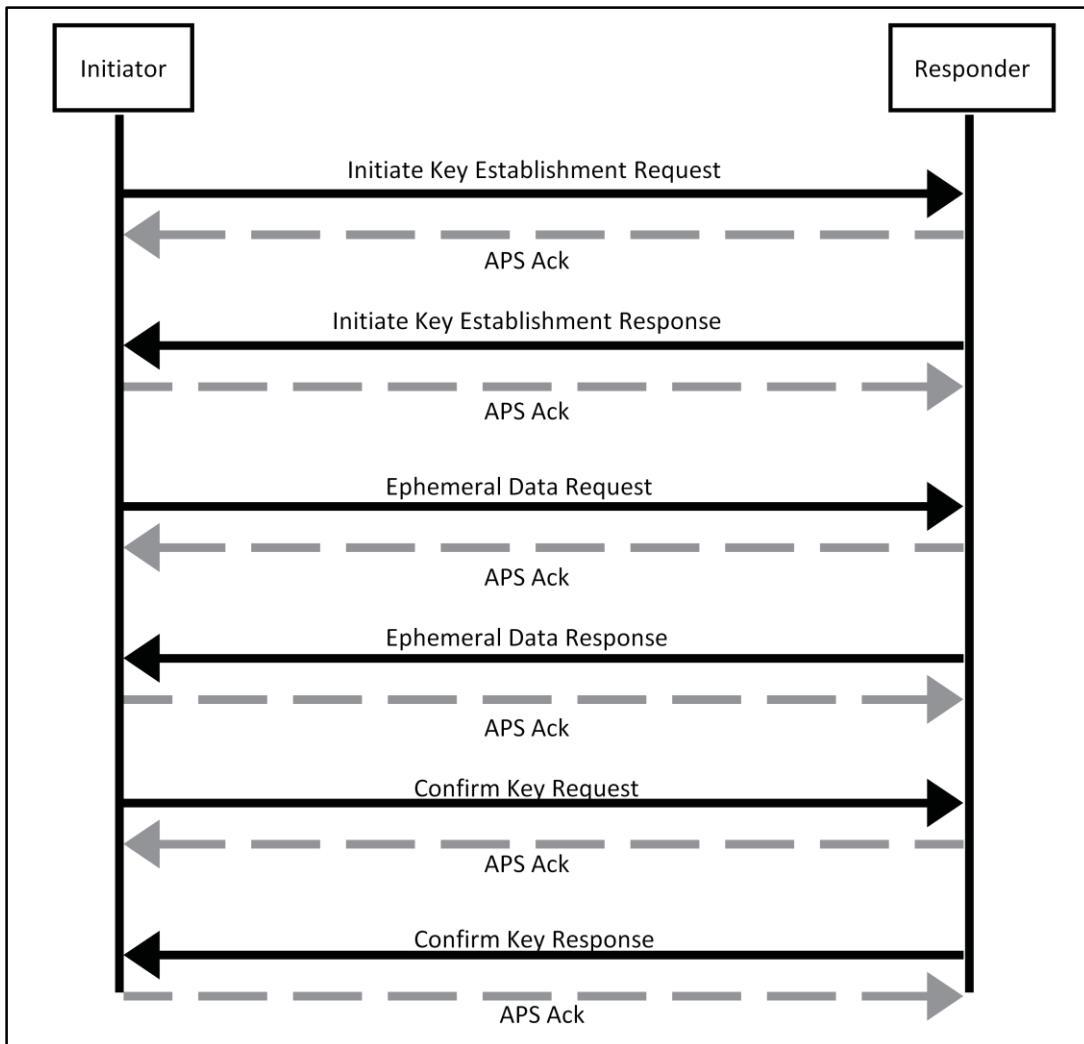
3538

3539 **Figure C-2– Typical Usage of the Key Establishment Cluster**

3540 **C.3.1 Key Establishment Cluster**

3541 **C.3.1.1 Overview**

3542 This cluster provides attributes and commands to perform mutual authentication and establish
3543 keys between two ZigBee devices. Figure C-3 depicts a diagram of a successful key
3544 establishment negotiation.



3545

3546

Figure C-3– Key Establishment Command Exchange

3547 As depicted above, all Key Establishment messages should be sent with APS retries enabled.
 3548 A failure to receive an ACK in a timely manner can be seen as a failure of key establishment.
 3549 No Terminate Key Establishment should be sent to the partner of device that has timed out the
 3550 operation.

3551 The initiator can initiate the key establishment with any active endpoint on the responder
 3552 device that supports the key establishment cluster. The endpoint can be either preconfigured or
 3553 discovered, for example, by using ZDO Match-Desc-req. A link key successfully established
 3554 using key establishment is valid for all endpoints on a particular device. The responder shall
 3555 respond to the initiator using the source endpoint of the initiator's messages as the destination
 3556 endpoint of the responder's messages.

3557 It is expected that the time it takes to perform the various cryptographic computations of
3558 the key establishment cluster may vary greatly based on the device. Therefore rather than set
3559 static timeouts, the *Initiate Key Establishment Request* and *Response* messages will contain
3560 approximate values for how long the device will take to generate the ephemeral data and how
3561 long the device will take to generate confirm key message.

3562 A device performing key establishment can use this information in order to choose a reasonable
3563 timeout for its partner during those operations. The timeout should also take into consideration
3564 the time it takes for a message to traverse the network including APS retries. A minimum
3565 transmission time of 2 seconds is recommended.

3566 For the *Initiate Key Establishment Response* message, it is recommended the initiator wait at
3567 least 2 seconds before timing out the operation. It is not expected that generating an *Initiate Key*
3568 *Establishment Response* will take significant time compared to generating the *Ephemeral Data*
3569 and *Confirm Key* messages.

3570 **C.3.1.1 Negotiating the Key Establishment Suite**

3571 Devices may support multiple cryptographic key establishment suites and therefore the client and
3572 server must agree on the suite that is to be used. Devices shall only advertise the suites that they
3573 support and have security credentials for.

3574 The client device is expected to negotiate the key establishment suite with the server, which will
3575 be used for the rest of the key establishment exchange. The initiating device (client) may perform
3576 a Read Attribute request on the *KeyEstablishmentSuite* attribute of the server. It will then
3577 compare its local value of the attribute to the server's value to determine the common set of
3578 suites that are supported by both. The client shall choose the common suite with the highest bit
3579 value and then send the *Initiate Key Establishment Request* message using that suite. If no
3580 common suites are supported, the device shall leave the network.

3581 **C.3.1.2 Server**

3582 **C.3.1.2.1 Dependencies**

3583 The Key Establishment server cluster has no dependencies.

3584 **C.3.1.2.2 Attributes**

3585 For convenience, the attributes defined in this specification are arranged into sets of related
3586 attributes; each set can contain up to 16 attributes. Attribute identifiers are encoded such that the
3587 most significant three nibbles specify the attribute set and the least significant nibble specifies
3588 the attribute within the set. The currently defined attribute sets are listed in Table C-2.

3589 **Table C-2– Key Establishment Attribute Sets**

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

3590

3591 C.3.1.2.2.1 Information

3592 The Information attribute set contains the attributes summarized in Table C.3.

3593

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

3594

3595 C.3.1.2.2.1.1 *KeyEstablishmentSuite* Attribute3596 The *KeyEstablishmentSuite* attribute is 16-bits in length and specifies ALL the cryptographic
3597 schemes for key establishment on the device. A device shall set the corresponding bit to 1 for
3598 every cryptographic scheme that it supports. All other cryptographic schemes and reserved bits
3599 shall be set to 0.3600 Although, for backwards compatibility, the Type cannot be changed, this 16-bit Enumeration
3601 should be treated as if it were a 16-bit BitMap.

3602

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

3603

3604 C.3.1.2.3 Commands Received3605 The server side of the key establishment cluster is capable of receiving the commands listed
3606 in Table C-5.

3607

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	

3608

3609 C.3.1.2.3.1 Initiate Key Establishment Request Command

3610 The *Initiate Key Establishment Request* command allows a device to initiate key establishment
 3611 with another device. The sender shall indicate the identity information and key establishment
 3612 protocol information that it wishes to use to the receiving device.

3613 C.3.1.2.3.1.1 *Payload Format*

3614 The *Initiate Key Establishment Request* command payload shall be formatted as illustrated in
 3615 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)

3616 Figure C-4– *Initiate Key Establishment Request Command Payload*

3617 **Key Establishment Suite:** This will be the type of *KeyEstablishmentSuite* that the initiator is
 3618 requesting for the Key Establishment Cluster. For ‘Crypto Suite 1’ this will be 0x0001. For
 3619 ‘Crypto Suite 2’ this will be 0x0002. Only one suite shall be indicated in the command.

3620 **Ephemeral Data Generate Time:** This value indicates approximately how long the initiator
 3621 device will take in seconds to generate the *Ephemeral Data Request* command. The valid
 3622 range is 0x00 to 0xFE.

3623
 3624 **Confirm Key Generate Time:** This value indicates approximately how long the initiator
 3625 device will take in seconds to generate the *Confirm Key Request* command. The valid range is
 3626 0x00 to 0xFE.
 3627

3628 **Identity field:** The identity field shall be the block of octets containing the implicit certificate
 3629 CERTU. For *KeyEstablishmentSuite* = 0x0001 (‘Crypto Suite 1’), the certificate is specified in

3630 sub-clause C.4.2.2. For *KeyEstablishmentSuite* = 0x0002 ('Crypto Suite 2') the certificate is
3631 specified in sub-clause C.4.2.3.

3632 **C.3.1.2.3.1.2 Effect on Receipt**

3633 If the device does not currently have the resources to respond to a key establishment
3634 request it shall send a *Terminate Key Establishment* command with the result value set to
3635 NO_RESOURCES and the Wait Time field shall be set to an approximation of the time that must
3636 pass before the device will have the resources to process a new Key Establishment Request.

3637 If the receiving device does not support the cryptographic suite specified in the message, it shall
3638 send a *Terminate Key Establishment* message with the status of UNSUPPORTED_SUITE.

3639 If the *KeyEstablishmentSuite* field of the message has more than a single bit selected in the
3640 bitmap, the receiving device shall send a *Terminate Key Establishment* message with the status of
3641 BAD_MESSAGE.

3642 The receiving device shall extract the Issuer field of the implicit certificate received in the
3643 message. It shall then examine all locally installed certificates using the same Cryptographic
3644 suite specified in the received message and compare the Issuer field contained within the
3645 certificate to the issuer within the received certificate. If no locally installed certificates match
3646 the issuer in the received certificate, the device shall send a *Terminate Key Establishment*
3647 command with the result set to UNKNOWN_ISSUER.

3648 If the implicit certificate received in the message is for the 'Crypto Suite 2' Cipher Suite, then
3649 the receiving device shall check the status of the *KeyUsage* field and, if the *Key Agreement*
3650 flag is NOT set, shall send a *Terminate Key Establishment* message with the status of
3651 INVALID_CERTIFICATE. The receiving device shall also check the *Type*, *Curve* and *Hash* fields of
3652 such a certificate, and send a *Terminate Key Establishment* message with the status of
3653 INVALID_CERTIFICATE if any of these fields contains an invalid value.

3654 If the device accepts the request it shall send an *Initiate Key Establishment Response*
3655 command containing its own identity information. It shall set the Key Establishment suite to the
3656 same value as in the received *Initiate Key Establishment Request* message. The identity
3657 information shall correspond to the same suite as specified in the Key Establishment suite. The
3658 device should verify the certificate belongs to the address that the device is communicating with.
3659 The binding between the identity of the communicating device and its address is verifiable using
3660 an out-of-band method.

3661 For all future server messages within the current key establishment negotiation, the Key
3662 Establishment suite value received in this message shall be utilized. If the client receives a
3663 *Terminate Key Establishment* message, or times out the operation, the key establishment suite
3664 value must be renegotiated.

3665 **C.3.1.2.3.2 Ephemeral Data Request Command**

3666 The *Ephemeral Data Request* command allows a device to communicate its ephemeral data
 3667 to another device and request that the device send back its own ephemeral data.

3668 **C.3.1.2.3.2.1 Payload Format**

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

3669 **Figure C-5– Ephemeral Data Request Command Payload**

3670 **C.3.1.2.3.2.2 Effect on Receipt**

3671 If the device is not currently in the middle of negotiating Key Establishment with the sending
 3672 device when it receives this message, it shall send back a Terminate Key Establishment
 3673 message with a result of BAD_MESSAGE. If the device is in the middle of Key Establishment
 3674 with the sender but did not receive this message in response to an *Initiate Key Establishment*
 3675 *Response* command, it shall send back a Terminate Key Establishment message with a result of
 3676 BAD_MESSAGE. If the device can process the request it shall respond by generating its
 3677 own ephemeral data and sending an *Ephemeral Data Response* command containing that value.

3678 The length of the frame shall correlate to the current key establishment suite that has been negotiated by
 3679 the client and server (refer to Table C-14 for relevant sizes). If the data is shorter than the expected
 3680 length according to the cryptographic suite, the responder shall send back a Terminate Key
 3681 Establishment message with a result of BAD_MESSAGE.

3682 **C.3.1.2.3.3 Confirm Key Request Command**

3683 The *Confirm Key Request* command allows the initiator sending device to confirm the key
 3684 established with the responder receiving device based on performing a cryptographic hash
 3685 using part of the generated keying material and the identities and ephemeral data of both parties.

3686 **C.3.1.2.3.3.1 Payload Format**

3687 The *Confirm KeyRequest* command payload shall be formatted as illustrated in Figure C-6.

3688

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

3689 **Figure C-6– Confirm Key Request Command Payload**

3690 **Secure Message Authentication Code field:** The Secure Message Authentication Code field
 3691 shall be the octet representation of *MACU* as specified in sub-clause C.4.2.

3692 **C.3.1.2.3.3.2 Effect on Receipt**

3693 If the device is not currently in the middle of negotiating Key Establishment with the sending
 3694 device when it receives this message, it shall send back a Terminate Key Establishment
 3695 message with a result of BAD_MESSAGE. If the device is in the middle of Key Establishment
 3696 with the sender but did not receive this message in response to an *Ephemeral Data Response*
 3697 command, it shall send back a Terminate Key Establishment message with a result of
 3698 BAD_MESSAGE.

3699 On receipt of the *Confirm Key Request* command the responder device shall compare the
 3700 received MACU value with its own reconstructed version of MACU. If the two match the
 3701 responder shall send back MACV by generating an appropriate *Confirm Key Response*
 3702 command. If the two do not match, the responder shall send back a Terminate Key
 3703 Establishment with a result of BAD KEY_CONFIRM and terminate the key establishment.

3704 **C.3.1.2.3.4 Terminate Key Establishment Command**

3705 The *Terminate Key Establishment* command may be sent by either the initiator or responder to
 3706 indicate a failure in the key establishment exchange.

3707 **C.3.1.2.3.4.1 Payload Format**

3708 The *Terminate Key Establishment* command payload shall be formatted as illustrated in Figure
 3709 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

3710 **Figure C-7– Terminate Key Establishment Command Payload**

3711 **Status Field:** The Status field shall be one of the error codes in Table C-6.

3712 **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 Key Agreement flag is not set in the <i>KeyUsage</i> field of the received 'Crypto Suite 2' certificate
	0x07 - 0xFF	Reserved

- 3713
- 3714 **Wait Time:** This value indicates the minimum amount of time in seconds the initiator device
- 3715 should wait before trying to initiate key establishment again. The valid range is 0x00 to 0xFE.
- 3716 **KeyEstablishmentSuite:** This value will be set the value of the *KeyEstablishmentSuite* attribute.
- 3717 It indicates the list of key exchange methods that the device supports.
- 3718 **C.3.1.2.3.4.2 Effect on Receipt**
- 3719 On receipt of the *Terminate Key Establishment* command the device shall terminate key
- 3720 establishment with the sender. If the device receives a status of BAD_MESSAGE or
- 3721 NO_RESOURCES it shall wait at least the time specified in the Wait Time field before trying to
- 3722 re-initiate Key Establishment with the device.
- 3723 If the device receives a status of UNSUPPORTED_SUITE it should examine the
- 3724 KeyEstablishmentSuite field to determine if another suite can be used that is supported by
- 3725 the partner device. It may re-initiate key establishment using that one of the supported suites
- 3726 after waiting the amount of time specified in the Wait Time field. If the device does not
- 3727 support any of the types in the KeyEstablishmentSuite field, it should not attempt key
- 3728 establishment again with that device.
- 3729 If the device receives a status of UNKNOWN_ISSUER or BAD_KEY_CONFIRM the device
- 3730 should not attempt key establishment again with the device, as it is unlikely that another attempt
- 3731 will be successful.
- 3732 **C.3.1.2.4 Commands Generated**
- 3733 The server generates the commands detailed in sub-clause C.3.1.3.3, as well as those used for
- 3734 reading and writing attributes.

3735 C.3.1.3 Client

3736 C.3.1.3.1 **Dependencies**

3737 The Key Establishment client cluster has no dependencies.

3738 C.3.1.3.2 **Attributes**

3739 For convenience, the attributes defined in this specification are arranged into sets of related
3740 attributes; each set can contain up to 16 attributes. Attribute identifiers are encoded such that the
3741 most significant three nibbles specify the attribute set and the least significant nibble specifies the
3742 attribute within the set. The currently defined attribute sets are listed in Table C-7.

3743

3744

3745 **Table C-7– Key Establishment Attribute Sets**

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

3746

3747 C.3.1.3.2.1 **Information**

3748 The Information attribute set contains the attributes summarized in Table C-8.

3749

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

3750

3751

3752

3753 ***C.3.1.3.2.1.1 KeyEstablishmentSuite Attribute***

3754 The *KeyEstablishmentSuite* attribute is 16-bits in length and specifies ALL the cryptographic
3755 schemes for key establishment on the device. A device shall set the corresponding bit to 1 for
3756 every cryptographic scheme that is supports. All other cryptographic schemes and reserved bits
3757 shall be set to 0. This attribute shall be set to one of the non-reserved values listed in Table C-9.

3758 Although, for backwards compatibility, the Type cannot be changed, this 16-bit Enumeration
 3759 should be treated as if it were a 16-bit BitMap.

3760
 3761

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

3762

3763 C.3.1.3.3 Commands Received

3764 The client side of the Key Establishment cluster is capable of receiving the commands
 3765 listed in Table C-10.

3766

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	

3767

3768 C.3.1.3.3.1 Initiate Key Establishment Response Command

3769 The *Initiate Key Establishment Response* command allows a device to respond to a device
 3770 requesting the initiation of key establishment with it. The sender will transmit its identity
 3771 information and key establishment protocol information to the receiving device.

3772 C.3.1.3.3.1.1 Payload Format

3773 The *Initiate Key Establishment Response* command payload shall be formatted as illustrated in
 3774 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

3775
3776

3777 **Requested Key Establishment Suite:** This will be the type of *KeyEstablishmentSuite* that the
3778 initiator has requested be used for the key establishment exchange. The responder device shall set
3779 a single bit in the bitmask indicating that it has accepted the requested suite; all other bits shall be
3780 set to zero.

3781 **Ephemeral Data Generate Time:** This value indicates approximately how long in seconds the
3782 responder device takes to generate the Ephemeral Data Response message. The valid range is
3783 0x00 to 0xFE.

3784 **Confirm Key Generate Time:** This value indicates approximately how long the responder
3785 device will take in seconds to generate the Confirm Key Response message. The valid range
3786 is 0x00 to 0xFE.

3787 **Identity field:** The *Identity* field shall be the block of octets containing the implicit certificate
3788 CERTU. For *KeyEstablishmentSuite* = 0x0001 ('Crypto Suite 1'), the certificate is specified in
3789 sub-clause C.4.2.2. For *KeyEstablishmentSuite* = 0x0002 ('Crypto Suite 2'), the certificate is
3790 specified in sub-clause C.4.2.2.3.

3791 **C.3.1.3.3.1.2 Effect on Receipt**

3792 If the device is not currently in the middle of negotiating Key Establishment with the sending
3793 device when it receives this message, it shall send back a Terminate Key Establishment
3794 message with a result of BAD_MESSAGE. If the device is in the middle of Key Establishment
3795 with the sender but did not receive this message in response to an *Initiate Key Establishment*
3796 *Request* command, it shall send back a Terminate Key Establishment message with a result of
3797 BAD_MESSAGE.

3798 If the receiving device does not support the key establishment suite specified in the message, it
3799 shall send a *Terminate Key Establishment* message with the status of UNSUPPORTED_SUITE.

3800 If the *Requested Key Establishment Suite* field of the message has more than a single bit selected
3801 in the bitmap, the receiving device shall send a *Terminate Key Establishment* message with the
3802 status of BAD_MESSAGE.

3803 On receipt of this command the device shall check the Issuer field of the device's implicit
3804 certificate. If the Issuer field does not contain a value that corresponds to a known Certificate
3805 Authority, the device shall send a *Terminate Key Establishment* command with the status
3806 value set to UNKNOWN_ISSUER. If the device does not currently have the resources to
3807 respond to a key establishment request it shall send a *Terminate Key Establishment* command
3808 with the status value set to NO_RESOURCES and the Wait Time field shall be set to an

3809 approximation of the time that must pass before the device has the resources to process the
3810 request.

3811 The receiver shall verify that the *KeyEstablishmentSuite* in the *Initiate Key Establishment*
3812 *Response* matches the value that was sent in the *Initiate Key Establishment Request*. If the values
3813 do not match then the device shall send a *Terminate Key Establishment Request* with
3814 UNSUPPORTED_SUITE.

3815 If the implicit certificate received in the message is for the ‘Crypto Suite 2’ Cipher Suite, then
3816 the receiving device shall check the status of the *KeyUsage* field and, if the *Key Agreement*
3817 flag is NOT set, shall send a *Terminate Key Establishment* message with the status of
3818 INVALID_CERTIFICATE. The receiving device shall also check the *Type*, *Curve* and *Hash* fields of
3819 such a certificate, and send a *Terminate Key Establishment* message with the status of
3820 INVALID_CERTIFICATE if any of these fields contains an invalid value.

3821 If the device accepts the response it shall send an *Ephemeral Data Request* command. The device
3822 should verify the certificate belongs to the address that the device is communicating with. The
3823 binding between the identity of the communicating device and its address is verifiable using out-
3824 of-band method.

3825 For all future client messages within the current key establishment negotiation, the Key
3826 Establishment suite value received in this message shall be utilized. If the client receives a
3827 *Terminate Key Establishment* message, or times out the operation, the key establishment suite
3828 value must be renegotiated.

3829 **C.3.1.3.3.2 Ephemeral Data Response Command**

3830 The *Ephemeral Data Response* command allows a device to communicate its ephemeral data
3831 to another device that previously requested it.

3832 **C.3.1.3.3.2.1 Payload Format**

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

3833 **Figure C-9–Ephemeral Data Response Command Payload**

3834 **C.3.1.3.3.2.2 Effect on Receipt**

3835 If the device is not currently in the middle of negotiating Key Establishment with the sending device
3836 when it receives this message, it shall send back a *Terminate Key Establishment* message with a
3837 result of BAD_MESSAGE. If the device is in the middle of Key Establishment with the sender but did
3838 not receive this message in response to an *Ephemeral Data Request* command, it shall send back a
3839 *Terminate Key Establishment* message with a result of BAD_MESSAGE.

3840 The length of the frame shall correlate to the current key establishment suite that has been negotiated by
 3841 the client and server (refer to Table C-14 for relevant sizes). If the length of the Ephemeral Data is
 3842 shorter than the expected length according to the cryptographic suite, the responder shall send back a
 3843 Terminate Key Establishment message with a result of BAD_MESSAGE.

3844 On receipt of this command if the device can handle the request it shall perform key generation, key
 3845 derivation, and MAC generation. If successful it shall generate an appropriate *Confirm Key*
 3846 *Request* command, otherwise it shall generate a Terminate Key Establishment with a result value
 3847 of NO_RESOURCES.

3848 C.3.1.3.3.3 Confirm Key Response Command

3849 The *Confirm Key Response* command allows the responder to verify the initiator has derived the
 3850 same secret key. This is done by sending the initiator a cryptographic hash generated using the
 3851 keying material and the identities and ephemeral data of both parties.

3852 C.3.1.3.3.3.1 *Payload Format*

3853 The *Confirm Key Response* command payload shall be formatted as illustrated in Figure C-10.

3854

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

3855 **Figure C-10– Confirm Key Response Command Payload**

3856 **Secure Message Authentication Code field:** The Secure Message Authentication Code field
 3857 shall be the octet representation of *MACV* as specified in sub-clause C.4.2.

3858 C.3.1.3.3.3.2 *Effect on Receipt*

3859 If the device is not currently in the middle of negotiating Key Establishment with the sending
 3860 device when it receives this message, it shall send back a Terminate Key Establishment
 3861 message with a result of BAD_MESSAGE. If the device is in the middle of Key Establishment
 3862 with the sender but did not receive this message in response to an *Confirm Key Request*
 3863 command, it shall send back a Terminate Key Establishment message with a result of
 3864 BAD_MESSAGE.

3865 On receipt of the *Confirm Key Response* command the initiator device shall compare the
 3866 received *MACV* value with its own reconstructed version of the *MACV*. If the two match
 3867 then the initiator can consider the key establishment process to be successful. If the two do

3868 not match, the initiator should send a *Terminate Key Establishment* command with a result of
 3869 BAD_KEY_CONFIRM.

3870 C.3.1.3.3.4 Terminate Key Establishment Command

3871 The *Terminate Key Establishment* command may be sent by either the initiator or responder to
 3872 indicate a failure in the key establishment exchange.

3873 **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

3874 **Figure C-11– Terminate Key Establishment Command Payload**

3875

3876 **Status field:** The Status field shall be one of the following error codes.

3877

3878

3879

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 Key Agreement flag is not set in the <i>KeyUsage</i> field of the received 'Crypto Suite 2' certificate
	0x07 - 0xFF	Reserved

3880

3881 **Wait Time:** This value indicates the minimum amount of time in seconds the initiator device

3882 should wait before trying to initiate key establishment again. The valid range is 0x00 to 0xFE.

3883 **KeyEstablishmentSuite:** This value will be set the value of the *KeyEstablishmentSuite* attribute.
3884 It indicates the list of key exchange methods that the device supports.

3885 **C.3.1.3.3.4.2 Effect on Receipt**

3886 On receipt of the *Terminate Key Establishment* command the device shall terminate key
3887 establishment with the sender. If the device receives a status of BAD_MESSAGE or
3888 NO_RESOURCES it shall wait at least the time specified in the Wait Time field before trying to
3889 re-initiate Key Establishment with the device.

3890 If the device receives a status of UNKNOWN_SUITE it should examine the
3891 *KeyEstablishmentSuite* field to determine if another suite can be used that is supported by the
3892 partner device. It may re-initiate key establishment using that one of the supported suites after
3893 waiting the amount of time specified in the Wait Time field. If the device does not support any of
3894 the types in the *KeyEstablishmentSuite* field, it should not attempt key establishment again with
3895 that device.

3896 If the device receives a status of UNKNOWN_ISSUER or *BAD_KEY_CONFIRM* the device
3897 should not attempt key establishment again with the device, as it is unlikely that another attempt
3898 will be successful.

3899 **C.3.1.3.4 Commands Generated**

3900 The client generates the commands detailed in sub-clause C.3.1.2.3, as well as those used for
3901 reading and writing attributes.

3902 **C.4 Application Implementation**

3903 **C.4.1 Network Security for Smart Energy Networks**

3904 The underlying network security for Smart Energy networks is assumed to be ZigBee
3905 Standard security using pre-configured link keys.

3906 A temporary link key for a joining device is produced by performing the cryptographic
3907 hash function on a random number assigned to the joining device (e.g. serial number) and the
3908 device identifier, which is the device's 64-bit IEEE address [B11].

3909 The joining device's assigned random number is then conveyed to the utility via an out-of-band
3910 mechanism (e.g. telephone call, or web site registration). The utility then commissions the
3911 energy service interface (ESI) at the premises where the joining device is by installing the
3912 temporary link key at the ESI on the back channel.

3913 When the joining device powers up, it will also create a temporary link key as above and
3914 therefore at the time of joining both the joining device and the ESI have the same temporary link
3915 key, which can be used to transport the network key securely to the joining device.

3916 At this point, the device will be considered joined and authenticated as far as network security
3917 is concerned. The secure communication cluster can now be invoked to replace the temporary
3918 link key with a more secure link key based on public key cryptography.

3919 **C.4.2 Certificate-Based Key Establishment**

3920 The Certificate-Based Key-Establishment (CBKE) solution uses public-key technology with
3921 digital certificates and root keys. Each device has a private key and a digital certificate that is
3922 signed by a Certificate Authority (CA).

3923 The digital certificate includes:

- 3924 • Reconstruction data for the device's public key
3925 • The device's extended 64-bit IEEE address
3926 • Profile specific information (e.g., the device class, network id, object type, validity date, etc.).

3927 Certificates provide a mechanism for cryptographically binding a public key to a device's
3928 identity and characteristics.

3929 Trust for a CBKE solution is established by provisioning a CA root key and a digital
3930 certificate to each device. A CA root key is the public key paired with the CA's private key. A
3931 CA uses its private key to sign digital certificates and the CA root key is used to verify these
3932 signatures. The trustworthiness of a public key is confirmed by verifying the CA's signature of
3933 the digital certificate. Certificates can be issued either by the device manufacturer, the device
3934 distributor, or the end customer. For example, in practical situations, the CA may be a computer
3935 (with appropriate key management software) that is kept physically secure at the end
3936 customer's facility or by a third-party.

3937 At the end of successful completion of the CBKE protocol the following security services are
3938 offered:

- 3939 • Both devices share a secret link key
3940 • Implicit Key Authentication: Both devices know with whom they share this link key.
3941 • Key Confirmation: Each device knows that the other device actually has computed the key
3942 correctly
3943 • No Unilateral Key Control: No device has complete control over the shared link key that is
3944 established.

- 3945 • Perfect Forward Secrecy: if the private key gets compromised none of future and past
3946 communications are exposed

- 3947 • Known Key Security resilience: Each shared link key created per session is unique

3948 **C.4.2.1 Notation and Representation**

3949 **C.4.2.1.1 Strings and String Operations**

3950 A string is a sequence of symbols over a specific set (e.g., the binary alphabet {0,1} or the set
3951 of all octets). The length of a string is the number of symbols it contains (over the same
3952 alphabet). The right-concatenation of two strings x and y of length m and n respectively
3953 (notation: $x // y$), is the string z of length $m+n$ that coincides with x on its leftmost m symbols
3954 and with y on its rightmost n symbols. An octet is a bit string of length 8.

3955 **C.4.2.1.2 Integers and their Representation**

3956 Throughout this specification, the representation of integers as bit strings or octet strings shall be
3957 fixed. All integers shall be represented as binary strings in most-significant-bit first order and
3958 as octet strings in most-significant-octet first order. This representation conforms to the
3959 convention in Section 2.3 of SEC1 [B18].

3960 **C.4.2.1.3 Entities**

3961 Throughout this specification, each entity shall be a DEV and shall be uniquely identified by
3962 its 64-bit IEEE device address [B11]. The parameter *entlen* shall have the integer value 64.

3963 **C.4.2.2 Cryptographic Suite 1 Building Blocks**

3964 The following cryptographic primitives and data elements are defined for use with the CBKE
3965 ‘Crypto Suite 1’ Cipher suite protocol specified in this document.

3966 **C.4.2.2.1 Elliptic-Curve Domain Parameters**

3967 The elliptic curve domain parameters used by this Cryptographic suite shall be those for the
3968 curve ‘sect163k1’ as specified in section 3.4.1 of SEC2 [B19].

3969 All elliptic-curve points (and operations in this section) used by the ‘Crypto Suite 1’ Cipher Suite
3970 shall be (performed) on this curve.

3971 **C.4.2.2.2 Elliptic-Curve Point Representation**

3972 All elliptic-curve points in the Cryptographic Suite 1 shall be represented as point compressed
3973 octet strings as specified in sections 2.3.3 and 2.3.4 of SEC1 [B18]. Thus, each elliptic-curve
3974 point Cryptographic Suite 1 can be represented in 22 bytes.

3975 C.4.2.2.3 **Elliptic-Curve Key Pair**

3976 An elliptic-curve-key pair consists of an integer d and a point Q on the curve determined by
3977 multiplying the generating point G of the curve by this integer (i.e., $Q=dG$) as specified in section
3978 3.2.1 of SEC1 [B18]. Here, Q is called the public key, whereas d is called the private key; the
3979 pair (d, Q) is called the key pair. Each private key shall be represented as specified in section
3980 2.3.7 of SEC1 [B18]. Each public key shall be represented as defined in sub-clause C.4.2.1.2 of
3981 this document.

3982 C.4.2.2.4 **ECC Implicit Certificates**

3983 The exact format of the 48-byte implicit certificate IC_U used with CBKE scheme shall be
3984 specified as follows:

3985
$$IC_U = PublicReconstrKey \parallel Subject \parallel Issuer \parallel ProfileAttributeData$$

3986 Where,

3987 1 *PublicReconstrKey*: the 22-byte representation of the public-key reconstruction data BEU as
3988 specified in the implicit certificate generation protocol, which is an elliptic-curve point as
3989 specified in sub-clause C.4.2.2.2 (see SEC4 [B18]);

3990 2 *Subject*: the 8-byte identifier of the entity U that is bound to the public-key reconstruction
3991 data BEU during execution of the implicit certificate generation protocol (i.e., the extended,
3992 64-bit IEEE 802.15.4 address [B11] of the device that purportedly owns the private key
3993 corresponding to the public key that can be reconstructed with *PublicReconstrKey*);

3994 3 *Issuer*: the 8-byte identifier of the CA that creates the implicit certificate during the execution
3995 of the implicit certificate generation protocol (the so-called Certificate Authority).

3996 4 *ProfileAttributeData*: the 10-byte sequence of octets that can be used by a ZigBee profile for
3997 any purpose. The first two bytes of this sequence is reserved as a profile identifier, which
3998 must be defined by another ZigBee standard.

3999 5 The string I_U as specified in Step 6 of the actions of the CA in the implicit certificate
4000 generation protocol (see section SEC4 [B22]) shall be the concatenation of the *Subject*,
4001 *Issuer*, and *ProfileAttributeData*:

4002
$$I_U = Subject \parallel Issuer \parallel ProfileAttributeData$$

4003 C.4.2.2.5 **Block-Cipher**

4004 The block-cipher used in this specification shall be the Advanced Encryption Standard AES-
4005 128, as specified in FIPS Pub 197 [B16]. This block-cipher has a key size that is equal to the
4006 block size, in bits, i.e., $keylen=128$.

4007 C.4.2.2.6 **Cryptographic Hash Function**

4008 The cryptographic hash function used in this specification shall be the blockcipher based
4009 cryptographic hash function specified in Annex B.6 in [B3], with the following
4010 instantiations:

4011 1 Each entity shall use the block-cipher E as specified in sub-clause B.1.1 in [B3].

4012 2 All integers and octets shall be represented as specified in sub-clause C.4.2.1.

4013 The Matyas-Meyer-Oseas hash function (specified in Annex B.6 in [B3]) has a message digest
4014 size *hashlen* that is equal to the block size, in bits, of the established blockcipher.

4015 C.4.2.2.7 **Keyed Hash Function for Message Authentication**

4016 The keyed hash message authentication code (HMAC) used in this specification shall be
4017 HMAC, as specified in the FIPS Pub 198 [B17] with the following instantiations:

4018 1 Each entity shall use the cryptographic hash *H* function as specified in sub- clause C.4.2.2.6;

4019 2 The block size *B* shall have the integer value 16 (this block size specifies the length of the
4020 data integrity key, in bytes, that is used by the keyed hash function, i.e., it uses a 128-bit data
4021 integrity key). This is also *MacKeyLen*, the length of *MacKey*.

4022 3 The output size *HMAClen* of the HMAC function shall have the same integer value as the
4023 message digest parameter *hashlen* as specified in sub- clause C.4.2.2.6.

4024 C.4.2.2.8 **Derived Shared Secret**

4025 The derived shared secret *KeyData* is the output of the key establishment. *KeyData* shall
4026 have length *KeyDataLen* of 128 bits.

4027 C.4.2.3 **Cryptographic Suite 2 Building Blocks**

4028 The elliptic curve domain parameters used by this Cipher suite shall be those for the curve
4029 “sect283k1” as specified in section 3.4.1 of SEC2 [B23].

4030 All elliptic-curve points (and operations in this section) used by the ‘Crypto Suite 2’ Cipher Suite
4031 shall be (performed) on this curve.

4032 C.4.2.3.1 **Elliptic-Curve Point Representation**

4033 All elliptic-curve points in the ‘Crypto Suite 2’ Cipher Suite shall be represented as point
4034 compressed octet strings as specified in sections 2.3.3 and 2.3.4 of SEC1 [B18]. Thus, each
4035 elliptic-curve point can be represented in 37 bytes.

4036 C.4.2.3.2 **Elliptic-Curve Key Pair**

4037 An elliptic-curve-key pair consists of an integer d and a point Q on the curve determined by
4038 multiplying the generating point G of the curve by this integer (i.e., $Q=dG$) as specified in section
4039 3.2.1 of SEC1 [B18]. Here, Q is called the public key, whereas d is called the private key; the
4040 pair (d, Q) is called the key pair. Each private key shall be represented as specified in section
4041 2.3.7 of SEC1 [B18]. Each public key shall be represented as defined in sub-clause C.4.2.1.2 of
4042 this document.

4043 C.4.2.3.3 **ECC Implicit Certificates**

4044 The exact format of the Cryptographic Suite 2 74-byte implicit certificate IC_U used with CBKE
4045 scheme follows the definitions given in SEC 4 [19] for the minimal encoding scheme (MES) and
4046 shall be specified as follows:

4047 $IC_U = Type \parallel SerialNo \parallel Curve \parallel Hash \parallel Issuer \parallel ValidFrom \parallel ValidTo \parallel Subject \parallel KeyUsage \parallel PublicReconstrKey$

4048 where

4049 1 ***Type***: is a 1-byte enumeration indicating whether the implicit certificate contains extensions.
4050 For the ‘Crypto Suite 2’ Cipher Suite this shall be 0x00 indicating no extensions are used;

4051 2 ***SerialNo***: is an 8-byte representation of the certificate Serial Number;

4052 3 ***Curve***: is a 1-byte elliptic curve identifier. For the ‘Crypto Suite 2’ Cipher Suite this shall be
4053 0x0D indicating the sect283k1 curve is used;

4054 4 ***Hash***: is a 1-byte hash identifier. For the ‘Crypto Suite 2’ Cipher Suite, this shall be 0x08 indicating
4055 that AES-MMO is used;

4056 5 ***Issuer***: the 8-byte address of the CA that creates the implicit certificate during the execution of
4057 the implicit certificate generation protocol (the Certificate Authority);

4058 6 ***ValidFrom***: the 5-byte Unix time from which the certificate is valid (this signed 40-bit
4059 integer matches that defined in SEC4 [B18]). For conversion between Unix and ZigBee
4060 time, the Zigbee Epoch (January 1, 2000) equates to 946,684,800 seconds in Unix time.
4061 *NOTE that this field is currently reserved and should be set to a default value of 0;*

4062 7 ***ValidTo***: a 4-byte number giving the seconds from the *ValidFrom* time for which the
4063 certificate is considered valid. A number less than 0xFFFFFFFF gives the number in seconds
4064 while 0xFFFFFFFF indicates an infinite number of seconds;

4065 8 ***Subject***: the 8-byte identifier of the entity U that is bound to the public-key reconstruction
4066 data BEU during execution of the implicit certificate generation protocol (i.e., the extended,
4067 64-bit IEEE 802.15.4 address [B11] of the device that purportedly owns the private key
4068 corresponding to the public key that can be reconstructed with *PublicReconstrKey*);

- 4069 9 *KeyUsage*: 1-byte identifier indicating the key usage. The complete bit string is defined in
 4070 SEC4 [B18], the bits relevant to the ‘Crypto Suite 2’ Cipher Suite are:-

4071 **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

4072
 4073 For usage of the ‘Crypto Suite 2’ Cipher Suite for Key Establishment, bit 3 shall be set;

- 4074 10 *PublicReconstrKey*: the 37-byte representation of the public-key reconstruction data BEU as
 4075 specified in the implicit certificate generation protocol, which is an elliptic-curve point as
 4076 specified in sub-clause C.4.2.2.2 (see SEC4 [B18]).

4077
 4078 The specification for ICu is further summarized in the following tabular form:

4079 **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.

- 4080
 4081 Note that the 74-byte certificate will necessitate the use of fragmentation with associated
 4082 commands.

- 4083
- 4084 C.4.2.3.4 **Block-Cipher**
4085 Refer to section C.4.2.2.5 for definition.
- 4086 C.4.2.3.5 **Cryptographic Hash Function**
4087 Refer to section C.4.2.2.6 for definition.
- 4088 C.4.2.3.6 **Keyed Hash Function for Message Authentication**
4089 Refer to section C.4.2.2.7 for definition.
- 4090 C.4.2.3.7 **Derived Shared Secret**
4091 Refer to section C.4.2.2.8 for definition.
4092
- 4093 **C.4.2.4 Certificate-Based Key-Establishment**
- 4094 The CBKE method is used when the authenticity of both parties involved has not been
4095 established and where implicit authentication of both parties is required prior to key agreement.
- 4096 The CBKE protocol has an identical structure to the PKKE protocol, except that implicit
4097 certificates are used rather than manual certificates. The implicit certificate protocol used with
4098 CBKE shall be the implicit certificate scheme with associated implicit certificate generation
4099 scheme and implicit certificate processing transformation as specified in SEC4 [B18],
4100 with the following instantiations:
- 4101 1 Each entity shall be a DEV;
- 4102 2 Each entity's identifier shall be its 64-bit device address [B11]; the parameter *entlen* shall
4103 have the integer value 64;
- 4104 3 Each entity shall use the cryptographic hash function as specified in sub-clause C.4.2.2.6;
- 4105 The following additional information shall have been unambiguously established between
4106 devices operating the implicit certificate scheme:
- 4107 1 Each entity shall have obtained information regarding the infrastructure that will be used for
4108 the operation of the implicit certificate scheme - including a certificate format and certificate
4109 generation and processing rules (see SEC4 [B18]);
- 4110 2 Each entity shall have access to an authentic copy of the elliptic-curve public keys of one or
4111 more certificate authorities that act as CA for the implicit certificate scheme (SEC4 [B18]).

- 4112 The methods by which this information is to be established are outside the scope of this
 4113 standard.
- 4114 The methods used during the CBKE protocol are described below. The parameters used by these
 4115 methods are described in Table C-14.

4116 **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 $(02\ 16\ //\ ID_U\ //\ ID_V\ //\ QE_U\ //\ QE_V)$ 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 $(03\ 16\ //\ ID_V\ //\ ID_U\ //\ QE_V\ //\ QE_U)$ 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 QE_V and QE_U 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]).

4117

4118 C.4.2.4.1 **Exchange Ephemeral Data**

4119 C.4.2.4.1.1 **Initiator**

4120 The initiator device's implicit certificate $CERT_U$ and a newly generated ephemeral public key Q_{EU} are transferred to the responder device using the *Initiate Key Establishment* command
4121 via the Key Establishment Cluster Client.
4122

4123 C.4.2.4.1.2 **Responder**

4124 The responder device's implicit certificate $CERT_V$ and a newly generated ephemeral
4125 public key Q_{EV} are transferred to the initiator device using the *Initiate Key Establishment*
4126 response command via the Key Establishment Cluster Server.

4127 C.4.2.4.2 **Validate Implicit Certificates**

4128 C.4.2.4.2.1 **Initiator**

4129 The initiator device's Key Establishment Cluster Client processes the *Initiate Key Establishment*
4130 response command. The initiator device examines $CERT_V$ (formatted as IC_V as described in sub-
4131 clause C.4.2.2.4), confirms that the *Subject* identifier is the purported owner of the certificate,
4132 and runs the certificate processing steps described in section SEC4 [B19].

4133 C.4.2.4.2.2 **Responder**

4134 The responder device's Key Establishment Cluster Server processes the *Initiate Key*
4135 *Establishment* command. The responder device examines $CERT_U$ (formatted as IC_U as described
4136 in sub-clause C.4.2.2.4), confirms that the *Subject* identifier is the purported owner of the
4137 certificate, and runs the certificate processing steps described in section SEC 4 [B19].

4138 C.4.2.4.3 **Derive Keying Material**

4139 C.4.2.4.3.1 **Initiator**

4140 The initiator performs the Elliptic Curve MQV scheme as specified in section 6.2 of SEC1 [B18]
4141 with the following instantiations:

- 4142 1 The elliptic curve domain parameters shall be as specified in sub- clause C.4.2.2.1;
- 4143 2 The KDF shall use the cryptographic hash function specified in sub- clause C.4.2.2.2;
- 4144 3 The static public key $Q_{1,U}$ shall be the static public key of the initiator;
- 4145 4 The ephemeral public key $Q_{2,U}$ shall be an ephemeral public key of the initiator generated as
4146 part of this transaction;

- 4147 **5** The static public key $Q_{1,V}$ shall be the static public key of the responder obtained from the
4148 responder's certificate communicated to the initiator by the responder;
- 4149 **6** The ephemeral public key $Q_{2,V}$ shall be based on the point-compressed octet string
4150 representation QE_V of an ephemeral key of the responder communicated to the initiator by
4151 the responder;
- 4152 **7** The KDF parameter *keydatalen* shall be $MacKeyLen + KeyDataLen$, where *MacKeyLen* is
4153 the length of *MacKey* and *KeyDataLen* is the length of *KeyData*;
- 4154 **8** The parameter *SharedInfo* shall be the empty string;
- 4155 The initiator device derives the keying material *MacKey* and *KeyData* from the output *K* as
4156 specified in section 3.6.1 of SEC1 [B18] by using *MacKey* as the leftmost *MacKeyLen* octets
4157 of *K* and *KeyData* as the rightmost *KeyDataLen* octets of *K*. *KeyData* is used subsequently as the
4158 shared secret and *MacKey* is used for key confirmation.

4159 C.4.2.4.3.2 Responder

- 4160 The responder performs the Elliptic Curve MQV scheme as specified in section 6.2 of SEC1
4161 [B18] with the following instantiations:
- 4162 **1** The elliptic curve domain parameters shall be as specified in sub- clause C.4.2.2.1;
- 4163 **2** The KDF shall use the cryptographic hash function specified in sub- clause C.4.2.2.2;
- 4164 **3** The static public key $Q_{1,U}$ shall be the static public key of the initiator obtained from the
4165 initiator's certificate communicated to the responder by the initiator;
- 4166 **4** The ephemeral public key $Q_{2,U}$ shall be based on the point-compressed octet string
4167 representation QE_U of an ephemeral key of the initiator communicated to the responder by
4168 the initiator;
- 4169 **5** The static public key $Q_{1,V}$ shall be the static public key of the responder;
- 4170 **6** The ephemeral public key $Q_{2,V}$ shall be an ephemeral public key of the responder generated
4171 as part of this transaction;
- 4172 **7** The KDF parameter *keydatalen* shall be $MacKeyLen + KeyDataLen$, where *MacKeyLen* is
4173 the length of *MacKey* and *KeyDataLen* is the length of *KeyData*;
- 4174 **8** The parameter *SharedInfo* shall be the empty string;

4175 The responder device derives the keying material *MacKey* and *KeyData* from the output *K* as
4176 specified in section 3.6.1 of SEC1 [B18] by using *MacKey* as the leftmost *MacKeyLen* octets
4177 of *K* and *KeyData* as the rightmost *KeyDataLen* octets of *K*. *KeyData* is used subsequently as the
4178 shared secret and *MacKey* is used for key confirmation.

4179 C.4.2.4.4 **Confirm Keys**

4180 C.4.2.4.4.1 Initiator

4181 The initiator device uses *MacKey* to compute its message authentication code *MACU* and
4182 sends it to the responder device by using the *Confirm Key* command via the Key Establishment
4183 Cluster Client.

4184 The initiator device uses *MacKey* to confirm the authenticity of the responder by calculating
4185 *MACV* and comparing it with that sent by the responder.

4186 C.4.2.4.4.2 Responder

4187 The responder device uses *MacKey* to compute its message authentication code *MACV* and
4188 sends it to the initiator device by using the *Confirm Key* response command via the Key
4189 Establishment Cluster Server.

4190 The responder device uses *MacKey* to confirm the authenticity of the initiator by calculating
4191 *MACU* and comparing it with that sent by the initiator.

4192 **C.5 Key Establishment Test Vectors for Cryptographic Suite 1**

4193 The following details the key establishment exchange data transformation and validation of
4194 test vectors for a pair of Smart Energy devices using Certificate based key exchange (CBKE)
4195 using Elliptical Curve Cryptography (ECC).

4196 **C.5.1 Preconfigured Data**

4197 Each device is expected to have been preinstalled with security information prior to initiating
4198 key establishment. The preinstalled data consists of the Certificate Authority's Public Key, a
4199 device specific certificate, and a device specific private key.

4200 **C.5.1.1 CA Public Key**

4201 The following is the Certificate Authority's Public Key.

4202 02 00 FD E8 A7 F3 D1 08
4203 42 24 96 2A 4E 7C 54 E6
4204 9A C3 F0 4D A6 B8

4205 C.5.1.2 Responder Data

4206 The following is the certificate for device 1. The device has an IEEE of (>)0000000000000001,
 4207 and will be the responder.

4208 03 04 5F DF C8 D8 5F FB

4209 8B 39 93 CB 72 DD CA A5

4210 5F 00 B3 E8 7D 6D 00 00

4211 00 00 00 00 00 01 54 45

4212 53 54 53 45 43 41 01 09

4213 00 06 00 00 00 00 00 00

4214 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

4215

4216 The private key for device 1 is as follows:

4217 00 b8 a9 00 fc ad eb ab

4218 bf a3 83 b5 40 fc e9 ed

4219 43 83 95 ea a7

4220 The public key for device 1 is as follows:

4221 03 02 90 a1 f5 c0 8d ad

4222 5f 29 45 e3 35 62 0c 7a

4223 98 fa c4 66 66 a1

4224 C.5.1.3 Initiator Data

4225 The following is the certificate for device 2. The device has an IEEE of (>)0000000000000002,
 4226 and will be the initiator.

4227 02 06 15 E0 7D 30 EC A2

4228 DA D5 80 02 E6 67 D9 4B

4229 C1 B4 22 39 83 07 00 00

4230 00 00 00 00 00 02 54 45

4231 53 54 53 45 43 41 01 09

4232 00 06 00 00 00 00 00 00

4233 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

4234

4235 The private key for device 2 is as follows:

4236 01 E9 DD B5 58 0C F7 2E

4237 CE 7F 21 5F 0A E5 94 E4

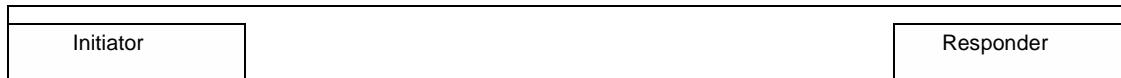
4238 8D F3 E7 FE E8

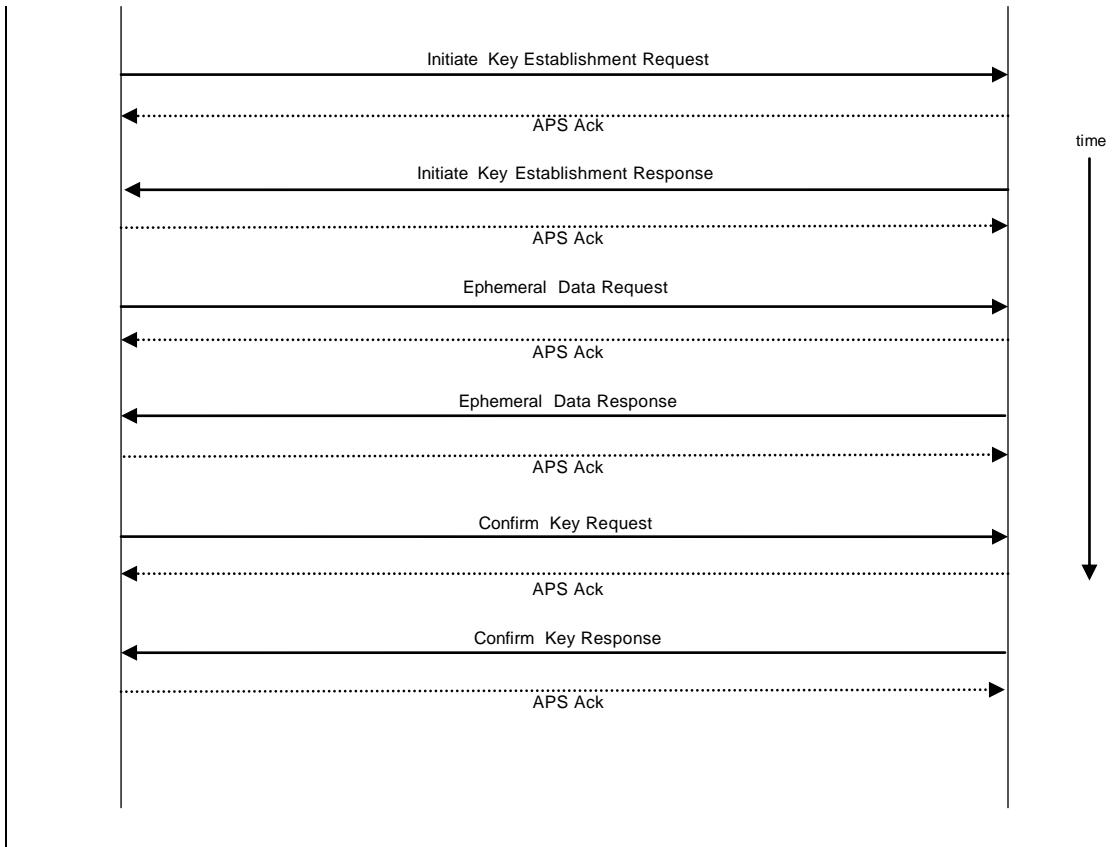
4239 The public key for device 2 is:

4240 03 02 5B BA 38 D0 C7 B5

4241 43 6B 68 DF 72 8F 09 3E

4242 7A 1D 6C 43 7E 6D

4243 **C.5.2 Key Establishment Messages**4244 The following is the basic flow of messages back and forth between the initiator and the
4245 responder performing key establishment using the Key Establishment Cluster.



4246

Figure C-12– Key Establishment Command Exchange**C.5.2.1 Initiate Key Establishment Request**

4248 The following is the APS message sent by the initiator (device 2) to the responder (device 1) for
4249 the initiate key establishment request.

4250 40 0A 00 08 09 01 0A 01
4251 01 00 00 01 00 03 06 02
4252 06 15 E0 7D 30 EC A2 DA
4253 D5 80 02 E6 67 D9 4B C1
4254 B4 22 39 83 07 00 00 00
4255 00 00 00 00 02 54 45 53
4256 54 53 45 43 41 01 09 00
4257 06 00 00 00 00 00 00

4258 APS Header

Frame Control	0x40
Destination Endpoint	0x0A
Cluster Identifier	0x0800

Profile ID	0x0109
Source Endpoint	0x0A
APS Counter	0x01

4259
4260**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

4261

C.5.2.2 Initiate Key Establishment Response

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

4265 40 0A 00 08 09 01 0A 01
 4266 09 00 00 01 00 03 06 03
 4267 04 5F DF C8 D8 5F FB 8B
 4268 39 93 CB 72 DD CA A5 5F
 4269 00 B3 E8 7D 6D 00 00 00
 4270 00 00 00 00 01 54 45 53
 4271 54 53 45 43 41 01 09 00
 4272 06 00 00 00 00 00 00 00

APS Header

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

4274
4275**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

4276

4277 **C.5.2.3 Ephemeral Data Request**4278 The following is the APS message sent by the initiator to the responder for the ephemeral data
4279 request.4280 40 0A 00 08 09 01 0A 02
4281 01 01 01 03 00 E1 17 C8
4282 6D 0E 7C D1 28 B2 F3 4E
4283 90 76 CF F2 4A F4 6D 72
4284 884285 **APS Header**

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

4286

4287 **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

4288

4289 **C.5.2.4 Ephemeral Data Response**4290 The following is the APS message sent by the responder to the initiator for the ephemeral data
4291 response.

4292 40 0A 00 08 09 01 0A 02

4293 09 01 01 03 06 AB 52 06
 4294 22 01 D9 95 B8 B8 59 1F
 4295 3F 08 6A 3A 2E 21 4D 84
 4296 5E

4297 APS Header

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

4298

4299 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

4300

4301 C.5.2.5 Confirm Key Request

4302 The following is the APS message sent by the initiator to the responder for the confirm key
 4303 request.

4304 40 0A 00 08 09 01 0A 03
 4305 01 02 02 B8 2F 1F 97 74
 4306 74 0C 32 F8 0F CF C3 92
 4307 1B 64 20

4308 APS Header

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

4309	APS Counter	0x02
------	-------------	------

4310 **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	

4311

4312 **C.5.2.6 Confirm Key Response**

4313 The following is the APS message sent by the responder to the initiator for the confirm key response.

4315 40 0A 00 08 09 01 0A 03

4316 09 02 02 79 D5 F2 AD 1C

4317 31 D4 D1 EE 7C B7 19 AC

4318 68 3C 3C

4319 **APS Header**

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

4320

4321 **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	

4322

4323 **C.5.3 Data Transformation**

4324 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

4325

4326 *Note: '||' stands for bitwise concatenation*

4327

4328 **C.5.3.1 ECMQV Primitives**4329 It is assumed that an ECC library is available for creating the shared secret given the local
4330 private key, local ephemeral public & private key, remote device's certificate, remote
4331 device's ephemeral public key, and the certificate authority's public key. Further it is assumed
4332 that this library has been separately validated with a set of ECC test vectors. Those test
4333 vectors are outside the scope of this document.4334 **C.5.3.2 Key Derivation Function (KDF)**4335 Once a shared secret (Z) is established, a transform is done to create a SMAC (Secure
4336 Message Authentication Code) and a shared ZigBee Key.

4337 C.5.3.3 Initiator Transform

4338 Upon receipt of the responder's ephemeral data response, the initiator has all the data necessary
 4339 to calculate the shared secret and derive the data for the confirm key request (SMAC).

4340 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

4341

4342 C.5.3.3.2 Step Summary

4343 1 Derive the Shared Secret using the ECMQV primitives

4344 a $Z = \text{ECC_GenerateSharedSecret}(\text{Private}(U), \text{E}(U), \text{E-P}(U), \text{Cert}(V), \text{E}(V), \text{CA})$

4345 2 Derive the Keying data

4346 a $\text{Hash-1} = Z \parallel 00\ 00\ 00\ 01 \parallel \text{SharedData}$

4347 b $\text{Hash-2} = Z \parallel 00\ 00\ 00\ 02 \parallel \text{SharedData}$

4348 3 Parse KeyingData as follows

4349 a $\text{MacKey} = \text{First 128 bits (Hash-1) of KeyingData}$

4350 b $\text{KeyData} = \text{Second 128 bits (Hash-2) of KeyingData}$

4351 4 Create MAC(U)

4352 a $\text{MAC}(U) = \text{MAC}(\text{MacKey}) \{ M(U) \parallel ID(U) \parallel ID(V) \parallel E(U) \parallel E(V) \}$

4353 5 Send MAC(U) to V.

4354 6 Receive MAC(V) from V.

4355 7 Calculate MAC(V)'

4356 a $\text{MAC}(V) = \text{MAC}(\text{MacKey}) \{ M(V) \parallel ID(V) \parallel ID(U) \parallel E(V) \parallel E(U) \}$

4357 8 Verify $\text{MAC}(V)'$ is the same as $\text{MAC}(V)$.4358 C.5.3.3.3 Detailed Steps

4359 1 Derive the Shared Secret using the ECMQV primitives

4360 a $Z = \text{ECC_GenerateSharedSecret}(\text{Private}(U), \text{E}(U), \text{E-P}(U), \text{Cert}(V), \text{E}(V), \text{CA})$

4361 00 E0 D2 C3 CC D5 C1 06 A8 9C 4F 6C C2 6A 5F 7E

4362 C9 DF 78 A7 BE

4363 2 Derive the Keying data

4364 a $\text{Hash-1} = Z \parallel 00\ 00\ 00\ 01 \parallel \text{SharedData}$

4365 **Concatenation**
 4366 00 E0 D2 C3 CC D5 C1 06 A8 9C 4F 6C C2 6A 5F 7E
 4367 C9 DF 78 A7 BE 00 00 00 01

4368 **Hash**
 4369 90 F9 67 B2 2C 83 57 C1 0C 1C 04 78 8D E9 E8 48

4370 b Hash-2 = Z || 00 00 00 02 || SharedData

4371 **Concatenation**
 4372 00 E0 D2 C3 CC D5 C1 06 A8 9C 4F 6C C2 6A 5F 7E
 4373 C9 DF 78 A7 BE 00 00 00 02

4374 **Hash**
 4375 86 D5 8A AA 99 8E 2F AE FA F9 FE F4 96 06 54 3A

4376 3 Parse KeyingData as follows

4377 a MacKey = First 128 bits (Hash-1) of KeyingData
 4378 b KeyData = Second 128 bits (Hash-2) of KeyingData

4380 4 Create MAC(U)

4381 a MAC(U) = MAC(MacKey) { M(U) || ID(U) || ID(V) || E(U) || E(V) }
 4382 **Concatenation**

4383 02 00 00 00 00 00 00 00 02 00 00 00 00 00 00 00 00 00
 4384 01 03 00 E1 17 C8 6D 0E 7C D1 28 B2 F3 4E 90 76
 4385 CF F2 4A F4 6D 72 88 03 06 AB 52 06 22 01 D9 95
 4386 B8 B8 59 1F 3F 08 6A 3A 2E 21 4D 84 5E 88 00 10

4387 **Hash**

4388 B8 2F 1F 97 74 74 0C 32 F8 0F CF C3 92 1B 64 20

4389 5 Send MAC(U) to V.

4390 6 Receive MAC(V) from V.

4391 7 Calculate MAC(V)'

4392 a MAC(V) = MAC(MacKey) { M(V) || ID(V) || ID(U) || E(V) || E(U) }
 4393 **Concatenation**

4394 03 00 00 00 00 00 00 00 00 01 00 00 00 00 00 00 00 00
 4395 02 03 06 AB 52 06 22 01 D9 95 B8 B8 59 1F 3F 08
 4396 6A 3A 2E 21 4D 84 5E 03 00 E1 17 C8 6D 0E 7C D1
 4397 28 B2 F3 4E 90 76 CF F2 4A F4 6D 72 88 88 00 10

4398 **Hash**

4399 79 D5 F2 AD 1C 31 D4 D1 EE 7C B7 19 AC 68 3C 3C

4400 8 Verify MAC(V)' is the same as MAC(V).

4401 C.5.3.4 Responder Transform

4402 Upon receipt of the initiator's confirm key request, the responder has all the data necessary to
 4403 calculate the shared secret, validate the initiator's confirm key message, and derive the data
 4404 for the confirm key response (SMAC).

4405 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

4406

4407 C.5.3.4.2 Step Summary

- 4408 1 Derive the Shared Secret using the ECMQV primitives
 - 4409 a $Z = \text{ECC_GenerateSharedSecret}(\text{Private}(V), E(V), E-P(V), \text{Cert}(U), E(U), CA)$
- 4410 2 Derive the Keying data
 - 4411 a $\text{Hash-1} = Z \parallel 00 00 00 01 \parallel \text{SharedData}$
 - 4412 b $\text{Hash-2} = Z \parallel 00 00 00 02 \parallel \text{SharedData}$
- 4413 3 Parse KeyingData as follows
 - 4414 a $\text{MacKey} = \text{First 128 bits (Hash-1) of KeyingData}$
 - 4415 b $\text{KeyData} = \text{Second 128 bits (Hash-2) of KeyingData}$
- 4416 4 Create MAC(V)
 - 4417 a $\text{MAC}(V) = \text{MAC}(\text{MacKey}) \{ M(V) \parallel ID(V) \parallel ID(U) \parallel E(V) \parallel E(U) \}$
- 4418 5 Calculate MAC(U)'
 - 4419 a $\text{MAC}(U) = \text{MAC}(\text{MacKey}) \{ M(U) \parallel ID(U) \parallel ID(V) \parallel E(U) \parallel E(V) \}$
- 4420 6 Verify $\text{MAC}(U)'$ is the same as $\text{MAC}(U)$.
- 4421 7 Send $\text{MAC}(V)$ to U.

4422 C.5.3.4.3 Detailed Steps

- 4423 1 Derive the Shared Secret using the ECMQV primitives
 - 4424 a $Z = \text{ECC_GenerateSharedSecret}(\text{Private}(U), E(U), E-P(U), \text{Cert}(V), E(V), CA)$

```
00 E0 D2 C3 CC D5 C1 06 A8 9C 4F 6C C2 6A 5F 7E
C9 DF 78 A7 BE
```
- 4427 2 Derive the Keying data
 - 4428 a $\text{Hash-1} = Z \parallel 00 00 00 01 \parallel \text{SharedData}$
 - 4429 **Concatenation**

4430 00 E0 D2 C3 CC D5 C1 06 A8 9C 4F 6C C2 6A 5F 7E
 4431 C9 DF 78 A7 BE 00 00 00 01

Hash

4433 90 F9 67 B2 2C 83 57 C1 0C 1C 04 78 8D E9 E8 48

4434 b Hash-2 = Z || 00 00 00 02 || SharedData

Concatenation

4436 00 E0 D2 C3 CC D5 C1 06 A8 9C 4F 6C C2 6A 5F 7E
 4437 C9 DF 78 A7 BE 00 00 00 02

Hash

4439 86 D5 8A AA 99 8E 2F AE FA F9 FE F4 96 06 54 3A

4440 3 Parse KeyingData as follows

4441 a MacKey = First 128 bits (Hash-1) of KeyingData

4442 b KeyData = Second 128 bits (Hash-2) of KeyingData

4443 4 Create MAC(V)

4444 a MAC(V) = MAC(MacKey) { M(V) || ID(V) || ID(U) || E(V) || E(U) }
Concatenation

4446 03 00 00 00 00 00 00 00 01 00 00 00 00 00 00 00 00
 4447 02 03 06 AB 52 06 22 01 D9 95 B8 B8 59 1F 3F 08
 4448 6A 3A 2E 21 4D 84 5E 03 00 E1 17 C8 6D 0E 7C D1
 4449 28 B2 F3 4E 90 76 CF F2 4A F4 6D 72 88 88 00 10

Hash

4451 79 D5 F2 AD 1C 31 D4 D1 EE 7C B7 19 AC 68 3C 3C

4452 5 Calculate MAC(V)'

4453 a MAC(U) = MAC(MacKey) { M(U) || ID(U) || ID(V) || E(U) || E(V) }
Concatenation

4455 02 00 00 00 00 00 00 00 00 02 00 00 00 00 00 00 00
 4456 01 03 00 E1 17 C8 6D 0E 7C D1 28 B2 F3 4E 90 76
 4457 CF F2 4A F4 6D 72 88 03 06 AB 52 06 22 01 D9 95
 4458 B8 B8 59 1F 3F 08 6A 3A 2E 21 4D 84 5E 88 00 10

Hash

4459 79 D5 F2 AD 1C 31 D4 D1 EE 7C B7 19 AC 68 3C 3C

4460 B8 2F 1F 97 74 74 0C 32 F8 0F CF C3 92 1B 64 20

4461 6 Verify MAC(V)' is the same as MAC(V).

4462 7 Send MAC(V) to U.

4463

C.6 Key Establishment Test Vectors for Cryptographic Suite 2

4465 *Note: The following information is provisional and yet to be verified. As a result, this information*
 4466 *may change before the suite becomes certifiable.* The following details the key establishment
 4467 exchange data transformation and validation of test vectors for a pair of Smart Energy

4468 devices using Certificate based key exchange (CBKE) using Elliptical Curve Cryptography
 4469 (ECC).

4470 C.6.1 Preconfigured Data

4471 Each device is expected to have been preinstalled with security information prior to initiating
 4472 key establishment. The preinstalled data consists of the Certificate Authority's Public Key, a
 4473 device specific certificate, and a device specific private key.

4474 C.6.1.1 CA Public Key

4475 The following is the Certificate Authority's Public Key:

4476 02 07 A4 45 02 2D 9F 39 f4 9B DC 38 38 00 26 A2
 4477 7A 9E 0A 17 99 31 3A B2 8C 5C 1A 1C 6B 60 51 54
 4478 DB 1D FF 67 52

4479 C.6.1.2 Responder Data

4480 The following is the certificate for device 1. The device has an EUI-64 address of
 4481 0A:0B:0C:0D:0E:0F:10:11, and will be the responder.

4482

4483 Certificate:

4484 00 26 22 A5 05 E8 93 8F 27 0D 08 11 12 13 14 15
 4485 16 17 18 00 52 92 A3 5B FF FF FF FF 0A 0B 0C 0D
 4486 0E 0F 10 11 88 03 03 B4 E9 DC 54 3A 64 33 3C 98
 4487 23 08 02 2B 54 E6 7E 2F 15 F5 32 55 1B 0A 11 E2
 4488 E2 C1 C1 D3 09 7A 43 24 E7 ED

4489

4490 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

4491

4492 The private key for device 1 is as follows:

4493 01 51 CD 0D BC B8 04 74 BF 7A C9 FE EB E3 9C 7A

4494 32 A6 35 18 93 8F CA 97 54 AA E1 32 BC 9C 73 BE
 4495 94 A7 E1 BE

4496 The public key for device 1 is as follows:

4497 02 02 F4 FA 2A 30 40 43 3C 68 20 29 9D 18 2A 10
 4498 42 E4 14 04 E3 37 C5 7F 47 71 6B 42 DF AF 97 0F
 4499 15 80 A0 4C 9B
 4500

4501 C.6.1.3 Initiator Data

4502 The following is the certificate for device 2. The device has an EUI-64 address of
 4503 0A:0B:0C:0D:0E:0F:10:12, and will be the initiator.

4504

4505 Certificate:

4506 00 84 A9 33 B3 7F 01 8D EC 0D 08 11 12 13 14 15
 4507 16 17 18 00 52 92 A3 8A FF FF FF FF 0A 0B 0C 0D
 4508 0E 0F 10 12 88 03 07 62 77 E2 F7 E2 25 2B 16 A0
 4509 E9 2B 6E 87 71 BB 3F 20 79 46 CB D4 A4 5D 9A 9D
 4510 F6 ED AB 8C 79 6A 48 E8 9D EC

4511

4512 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

4513

4514 The private key for device 2 is as follows:

4515 00 F2 56 1A DB 39 EF 49 C1 D6 2E F5 18 6C 6E 0C
 4516 15 8A 5A 45 BF CE 38 66 09 31 AC C3 69 45 92 D5
 4517 AC DE 90 06

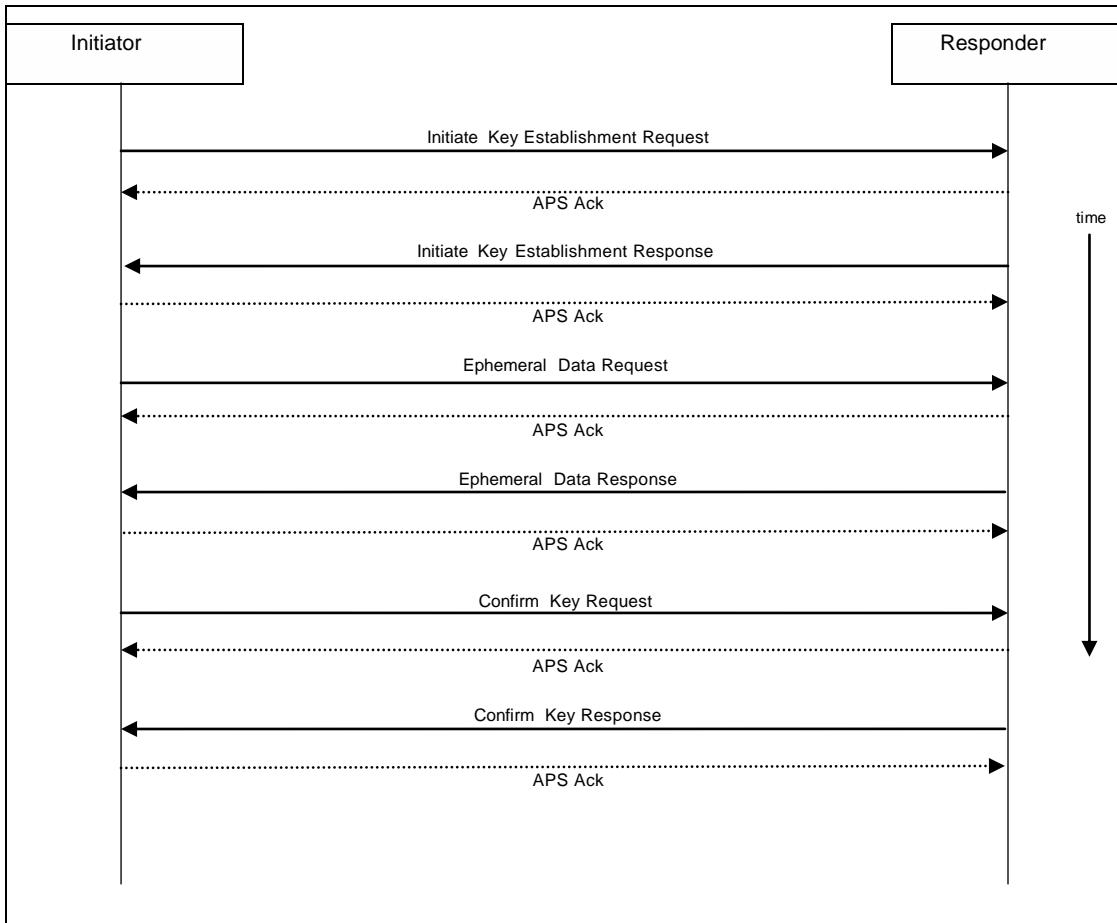
4518 The public key for device 2 is as follows:

4519 03 03 0E 56 F7 AD E8 66 E7 63 72 76 4B A2 0A 9F
 4520 F1 FE 4C AE 52 2F 94 83 9E 70 F2 AD FC 1C A3 E9
 4521 7F 4D DC AF 2E

4522

4523 C.6.2 Key Establishment Messages

4524 The following is the basic flow of messages back and forth between the initiator and the
 4525 responder performing key establishment using the Key Establishment Cluster.



4526

Figure C-13– Key Establishment Command Exchange

4527 C.6.2.1 Initiate Key Establishment Request

4528 The following is the APS message sent by the initiator (device 2) to the responder (device 1) for
 4529 the initiate key establishment request.

4530 40 0A 00 08 09 01 0A 01 01 00 00 02 00 03 06 00
 4531 84 A9 33 B3 7F 01 8D EC 0D 08 11 12 13 14 15 16
 4532 17 18 00 52 92 A3 8A FF FF FF FF 0A 0B 0C 0D 0E
 4533 0F 10 12 88 03 07 62 77 E2 F7 E2 25 2B 16 A0 E9
 4534 2B 6E 87 71 BB 3F 20 79 46 CB D4 A4 5D 9A 9D F6
 4535 ED AB 8C 79 6A 48 E8 9D EC

4536 APS Header

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

4537

4538

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

4539

4540 **C.6.2.2 Initiate Key Establishment Response**
 4541 The following is the APS message sent by the responder (device 1) to the initiator (device 2) for
 4542 the initiate key establishment response.

```

 4543 40 0A 00 08 09 01 0A 01 09 00 00 02 00 03 06 00
 4544 26 22 A5 05 E8 93 8F 27 0D 08 11 12 13 14 15 16
 4545 17 18 00 52 92 A3 5B FF FF FF FF 0A 0B 0C 0D 0E
 4546 0F 10 11 88 03 03 B4 E9 DC 54 3A 64 33 3C 98 23
 4547 08 02 2B 54 E6 7E 2F 15 F5 32 55 1B 0A 11 E2 E2
 4548 C1 C1 D3 09 7A 43 24 E7 ED
  
```

4549

APS Header

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

4550

4551

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

4552

4553 C.6.2.3 Ephemeral Data Request

4554 The following is the APS message sent by the initiator to the responder for the ephemeral data
4555 request.

4556 40 0A 00 08 09 01 0A 02 01 01 01 03 05 F3 39 4E
4557 15 68 06 60 EE CA A3 67 88 D9 B6 F3 12 B9 71 CE
4558 2C 96 17 57 0B F7 DF CD 21 C9 72 01 77 62 C3 32

4559 APS Header

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

4560

4561 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

4562

4563 C.6.2.4 Ephemeral Data Response

4564 The following is the APS message sent by the responder to the initiator for the ephemeral data
4565 response.

4566 40 0A 00 08 09 01 0A 02 09 01 01 03 00 9A 51 31
4567 CF 5B 92 A0 16 37 8C 0F 7F 28 4E CD 47 F9 40 10
4568 F8 75 D4 3B F1 E9 A6 54 74 AD BF C6 36 96 A9 30

4569 APS Header

Frame Control	0x40
---------------	------

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

4570

4571 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

4572

4573 C.6.2.5 Confirm Key Request

4574 The following is the APS message sent by the initiator to the responder for the confirm key
 4575 request.

4576 40 0A 00 08 09 01 0A 03
 4577 01 02 02 BF 7E 1A 26 D4
 4578 EF 70 38 B5 68 13 E4 65
 4579 A1 31 C9

4580

APS Header

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

4581

4582 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

4583

4584 **C.6.2.6 Confirm Key Response**

4585 The following is the APS message sent by the responder to the initiator for the confirm key
 4586 response.

4587 40 0A 00 08 09 01 0A 03
 4588 09 02 02 C5 B4 32 A9 99
 4589 5A 09 2F 44 49 F8 36 13
 4590 93 00 64

4591 **APS Header**

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

4592

4593 **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	

4594

4595 **C.6.3 Data Transformation**

4596 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

4597

4598 *Note: '||' stands for bitwise concatenation*4599 **C.6.3.1 ECMQV Primitives**

4600 It is assumed that an ECC library is available for creating the shared secret given the local
 4601 private key, local ephemeral public & private key, remote device's certificate, remote
 4602 device's ephemeral public key, and the certificate authority's public key. Further it is assumed
 4603 that this library has been separately validated with a set of ECC test vectors. Those test
 4604 vectors are outside the scope of this document.

4605 **C.6.3.2 Key Derivation Function (KDF)**

4606 Once a shared secret (Z) is established, a transform is done to create a SMAC (Secure
 4607 Message Authentication Code) and a shared ZigBee Key.

4608 **C.6.3.3 Initiator Transform**

4609 Upon receipt of the responder's ephemeral data response, the initiator has all the data necessary
 4610 to calculate the shared secret and derive the data for the confirm key request (SMAC).

4611 **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

4612

4613 **C.6.3.3.2 Step Summary**

- 4614 **1** Derive the Shared Secret using the ECMQV primitives
 4615 a Z = ECC_GenerateSharedSecret(Private(U), E(U), E-P(U), Cert(V), E(V), CA)
- 4616 **2** Derive the Keying data
 4617 a Hash-1 = Z || 00 00 00 01 || SharedData
 4618 b Hash-2 = Z || 00 00 00 02 || SharedData

- 4619 **3** Parse KeyingData as follows
 4620 a MacKey = First 128 bits (Hash-1) of KeyingData
 4621 b KeyData = Second 128 bits (Hash-2) of KeyingData
- 4622 **4** Create MAC(U)
 4623 a MAC(U) = MAC(MacKey) { M(U) || ID(U) || ID(V) || E(U) || E(V) }
- 4624 **5** Send MAC(U) to V.
- 4625 **6** Receive MAC(V) from V.
- 4626 **7** Calculate MAC(V)'
 4627 a MAC(V)' = MAC(MacKey) { M(V) || ID(V) || ID(U) || E(V) || E(U) }
- 4628 **8** Verify MAC(V)' is the same as MAC(V).
- 4629

4630 C.6.3.3.3 **Detailed Steps**

- 4631 **1** Derive the Shared Secret using the ECMQV primitives
 4632 a Z = ECC_GenerateSharedSecret(Private(U), E(U), E-P(U), Cert(V), E(V), CA)
 04 F7 72 4A 9A 77 B2 1D 27 47 CC EF 68 A4 57 E4
 52 46 C4 BE 9F 66 FD 94 25 22 7B CB 2C C5 18 0E
 A9 CC CB 9A
- 4636 **2** Derive the Keying data
 4637 a Hash-1 = Z || 00 00 00 01 || SharedData
 Concatenation
 04 F7 72 4A 9A 77 B2 1D 27 47 CC EF 68 A4 57 E4
 52 46 C4 BE 9F 66 FD 94 25 22 7B CB 2C C5 18 0E
 A9 CC CB 9A 00 00 01
- 4642 **Hash**
 4643 ED 38 0A 00 29 66 00 FB 6B 89 30 25 DE 5F D1 37
- 4644
- 4645 b Hash-2 = Z || 00 00 00 02 || SharedData
 Concatenation
 04 F7 72 4A 9A 77 B2 1D 27 47 CC EF 68 A4 57 E4
 52 46 C4 BE 9F 66 FD 94 25 22 7B CB 2C C5 18 0E
 A9 CC CB 9A 00 00 02
- 4650 **Hash**
 4651 AA 46 89 C7 0B E0 FA F0 C9 BE 53 4A BD 9F 4C DC
- 4652
- 4653 **3** Parse KeyingData as follows
 4654 a MacKey = First 128 bits (Hash-1) of KeyingData
 4655 b KeyData = Second 128 bits (Hash-2) of KeyingData

4656 **4 Create MAC(U)**

4657 a $\text{MAC}(U) = \text{MAC}(\text{MacKey}) \{ M(U) \parallel ID(U) \parallel ID(V) \parallel E(U) \parallel E(V) \}$

Concatenation

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

Hash

BF 7E 1A 26 D4 EF 70 38	B5 68 13 E4 65 A1 31 C9
-------------------------	-------------------------

4668 **5 Send MAC(U) to V.**

4669 **6 Receive MAC(V) from V.**

4670 **7 Calculate MAC(V)'**

4671 a $\text{MAC}(V) = \text{MAC}(\text{MacKey}) \{ M(V) \parallel ID(V) \parallel ID(U) \parallel E(V) \parallel E(U) \}$

Concatenation

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

Hash

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

4682 **8 Verify $\text{MAC}(V)'$ is the same as $\text{MAC}(V)$.**

4683

4684 **C.6.3.4 Responder Transform**

4685 Upon receipt of the initiator's confirm key request, the responder has all the data necessary to
 4686 calculate the shared secret, validate the initiator's confirm key message, and derive the data
 4687 for the confirm key response (SMAC).

4688 **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

4689

4690 C.6.3.4.2 **Step Summary**

4691 1 Derive the Shared Secret using the ECMQV primitives

4692 a $Z = \text{ECC_GenerateSharedSecret}(\text{Private}(V), E(V), E-P(V), \text{Cert}(U), E(U), CA)$

4693 2 Derive the Keying data

4694 a $\text{Hash-1} = Z \parallel 00\ 00\ 00\ 01 \parallel \text{SharedData}$ 4695 b $\text{Hash-2} = Z \parallel 00\ 00\ 00\ 02 \parallel \text{SharedData}$

4696 3 Parse KeyingData as follows

4697 a $\text{MacKey} = \text{First 128 bits (Hash-1) of KeyingData}$ 4698 b $\text{KeyData} = \text{Second 128 bits (Hash-2) of KeyingData}$

4699 4 Create MAC(V)

4700 a $\text{MAC}(V) = \text{MAC}(\text{MacKey}) \{ M(V) \parallel ID(V) \parallel ID(U) \parallel E(V) \parallel E(U) \}$

4701 5 Calculate MAC(U)'

4702 a $\text{MAC}(U) = \text{MAC}(\text{MacKey}) \{ M(U) \parallel ID(U) \parallel ID(V) \parallel E(U) \parallel E(V) \}$ 4703 6 Verify $\text{MAC}(U)'$ is the same as $\text{MAC}(U)$.

4704 7 Send MAC(V) to U

4705

4706 C.6.3.4.3 **Detailed Steps**

4707 1 Derive the Shared Secret using the ECMQV primitives

4708 a $Z = \text{ECC_GenerateSharedSecret}(\text{Private}(V), E(V), E-P(V), \text{Cert}(U), E(U), CA)$

4709 04 F7 72 4A 9A 77 B2 1D 27 47 CC EF 68 A4 57 E4

4710 52 46 C4 BE 9F 66 FD 94 25 22 7B CB 2C C5 18 0E

4711 A9 CC CB 9A

4712 2 Derive the Keying data

4713 a $\text{Hash-1} = Z \parallel 00\ 00\ 00\ 01 \parallel \text{SharedData}$ 4714 **Concatenation**

4715 04 F7 72 4A 9A 77 B2 1D 27 47 CC EF 68 A4 57 E4

4716 52 46 C4 BE 9F 66 FD 94 25 22 7B CB 2C C5 18 0E

4717 A9 CC CB 9A 00 00 00 01

4718 **Hash**

4719 ED 38 0A 00 29 66 00 FB 6B 89 30 25 DE 5F D1 37

4720

4721 b $\text{Hash-2} = Z \parallel 00\ 00\ 00\ 02 \parallel \text{SharedData}$ 4722 **Concatenation**

4723 04 F7 72 4A 9A 77 B2 1D 27 47 CC EF 68 A4 57 E4

4724 52 46 C4 BE 9F 66 FD 94 25 22 7B CB 2C C5 18 0E

4725 A9 CC CB 9A 00 00 00 02

4726 **Hash**
 4727 AA 46 89 C7 0B E0 FA F0 C9 BE 53 4A BD 9F 4C DC
 4728

- 4729 **3** Parse KeyingData as follows
 4730 a MacKey = First 128 bits (Hash-1) of KeyingData
 4731 b KeyData = Second 128 bits (Hash-2) of KeyingData

- 4732 **4** Create MAC(V)
 4733 a MAC(V) = MAC(MacKey) { M(V) || ID(V) || ID(U) || E(V) || E(U) }

4734 **Concatenation**
 4735 03 0a 0b 0c 0d 0e 0f 10 11 0a 0b 0c 0d 0e 0f 10
 4736 12 03 00 9A 51 31 CF 5B 92 A0 16 37 8C 0F 7F 28
 4737 4E CD 47 F9 40 10 F8 75 D4 3B F1 E9 A6 54 74 AD
 4738 BF C6 36 96 A9 30 03 05 F3 39 4E 15 68 06 60 EE
 4739 CA A3 67 88 D9 B6 F3 12 B9 71 CE 2C 96 17 57 0B
 4740 F7 DF CD 21 C9 72 01 77 62 C3 32

4741 **Hash**
 4742 C5 B4 32 A9 99 5A 09 2F 44 49 F8 36 13 93 00 64
 4743

- 4744 **5** Calculate MAC(U)'
 4745 a MAC(U)' = MAC(MacKey) { M(U) || ID(U) || ID(V) || E(U) || E(V) }
 4746 **Concatenation**
 4747 02 0a 0b 0c 0d 0e 0f 10 12 0a 0b 0c 0d 0e 0f 10
 4748 11 03 05 F3 39 4E 15 68 06 60 EE CA A3 67 88 D9
 4749 B6 F3 12 B9 71 CE 2C 96 17 57 0B F7 DF CD 21 C9
 4750 72 01 77 62 C3 32 03 00 9A 51 31 CF 5B 92 A0 16
 4751 37 8C 0F 7F 28 4E CD 47 F9 40 10 F8 75 D4 3B F1
 4752 E9 A6 54 74 AD BF C6 36 96 A9 30

4753 **Hash**
 4754 BF 7E 1A 26 D4 EF 70 38 B5 68 13 E4 65 A1 31 C9
 4755

- 4756 **6** Verify MAC(U)' is the same as MAC(U).

- 4757 **7** Send MAC(V) to U
 4758

4759

4760

Annex D SMART ENERGY CLUSTER DESCRIPTIONS

4761

4762 The candidate material in this annex describing the Smart Energy Clusters, when approved, will
4763 be merged into the Foundation document of the ZigBee Cluster Library (ZCL) by the Cluster
4764 Library Development Board.

4765

D.1 Annex Guidelines

4766

D.1.1 Client/Server Model Information

4767

4768 The ZigBee Cluster Library Specification is used as the guiding reference for defining the
4769 rule set in defining the Client/Server model for the Smart Energy Profile. Please note the
following items influence the further refinement of that definition:

4770

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

4773

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

4779

D.1.2 Interpretation of Reserved Field Values or Bitmaps

4780

4781 To support backwards compatibility, devices should ignore any values or bit settings for any
4782 reserved field values. If the field is necessary for interpretation or in conjunction with other fields
the whole message can be ignored.

4783

4784 To enable future growth and ensure backwards compatibility, any existing devices which
4785 encounter any fields applied after the end of a command shall treat them as reserved fields. The
4786 future addition of fields applied after the end of defined cluster commands are reserved
solely for ZigBee specifications, Manufacturers shall not add fields after the end of commands.

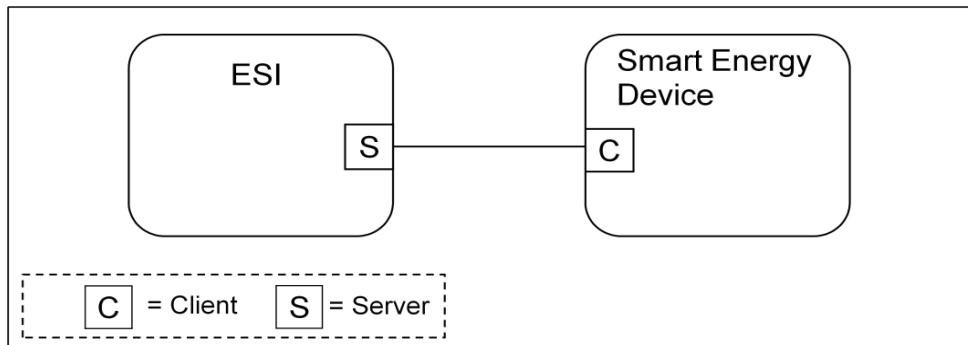
4787

4788 **D.2 Demand Response and Load Control Cluster**

4789 **D.2.1 Overview**

4790 This cluster provides an interface to the functionality of Smart Energy Demand Response and
4791 Load Control. Devices targeted by this cluster include thermostats and devices that support load
4792 control.

4793



4794
4795 **Figure D-1– Demand Response/Load Control Cluster Client Server Example**

4796 Please note the ESI is defined as the Server due to its role in acting as the proxy for upstream
4797 demand response/load control management systems and subsequent data stores.

4798 **D.2.2 Server**

4799 By default the ESI will be labeled as the Server side in the cluster descriptions, being able to
4800 initiate load control commands to other devices in the network.

4801 **D.2.2.1 Dependencies**

4802 A server device shall be capable of storing at least two load control events.

4803 Events carried using this cluster include a timestamp with the assumption that target devices
4804 maintain a real-time clock. Devices can acquire and synchronize their internal clocks with the
4805 ESI as described in sub-clause 5.12.1.1.

4806 If a device does not support a real-time clock, it is assumed the device will ignore all values within
4807 the Time field except the “Start Now” value.

4808 Additionally, for devices without a real-time clock, it is assumed those devices will utilize a
4809 method (i.e. ticks, countdowns, etc.) to approximate the correct duration period.

4810 **D.2.2.2 Attributes**

4811 There are no attributes for the Demand Response and Load Control Cluster server.

4812 **D.2.2.3 Commands Generated**

4813 The command IDs generated by the Demand Response and Load Control cluster server are
4814 listed in Table D-1.

4815 **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	

4816

4817 **D.2.2.3.1 Load Control Event Command**

4818 **D.2.2.3.1.1 Payload Format**

4819 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)

4820

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)

4821 **Figure D-2– Load Control Event Command Payload**

4822 Note: M = Mandatory field, O = Optional field. All fields must be present in the payload. Optional fields will be
4823 marked with specific values to indicate they are not being used.

4824 **D.2.2.3.1.1.1 Payload Details**

4825 **Issuer Event ID (mandatory):** Unique identifier generated by the Energy provider. The value of
 4826 this field allows matching of Event reports with a specific Demand Response and Load Control
 4827 event. The expected value contained in this field shall be a unique number managed by upstream
 4828 systems or a UTC based time stamp (UTCTime data type) identifying when the Load Control Event
 4829 was issued.

4830 **Device Class (mandatory):** Bit encoded field representing the Device Class to apply the
 4831 current Load Control Event. Each bit, if set individually or in combination, indicates the class
 4832 device(s) needing to participate in the event. (Note that the participating device may be
 4833 different than the controlling device. For instance, a thermostat may act on behalf of an HVAC
 4834 compressor or furnace and/or Strip Heat/Baseboard Heater and should take action on their
 4835 behalf, as the thermostat itself is not subject to load shed but controls devices that are subject to
 4836 load shed.) The encoding of this field is in Table D-2:

4837 **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

4838 Device manufacturers shall recognize the Device Class or set of Devices Classes that
 4839 corresponds to its functionality. For example, a thermostat (PCT) may react when Bit 0 is set
 4840 since it controls the HVAC and/or furnace. Another example is a device that acts like an EMS
 4841 where it controls exterior lights, interior lights, and simple misc. load control devices. In this
 4842 case the EMS would react when Bits 7, 8, or 9 are set individually or in combination.
 4843

4844 **Utility Enrollment Group (mandatory):** The Utility Enrollment Group field can be used in
 4845 conjunction with the Device Class bits. It provides a mechanism to direct Load Control Events
 4846 to groups of Devices. Example, by assigning two different groups relating to either Demand

- 4847 Response programs or geographic areas, Load Control Events can be further directed for a sub-
 4848 set of Device Classes (i.e. Device Class Bit 0 and Utility Enrollment Group #1 vs. Device
 4849 Class Bit0 and Utility Enrollment Group #2). 0x00 addresses all groups, and values 0x01 to
 4850 0xFF address individual groups that match. Please refer to sub-clause D.2.3.2.1 for further
 4851 details.
- 4852 If the Device Class and/or Utility Enrollment Group fields don't apply to your End Device, the
 4853 *Load Control Event* command shall be ignored by either dropping the message and not replying
 4854 at all or by sending back a Default Response message with a SUCCESS status code.
- 4855 **Start Time (mandatory):** UTC Timestamp representing when the event is scheduled to start. A
 4856 start time of 0x00000000 is a special time denoting "now." If the device would send an event
 4857 with a Start Time of now, adjust the Duration In Minutes field to correspond to the remainder of
 4858 the event.
- 4859 **Duration In Minutes (mandatory):** Duration of this event in number of minutes. Maximum
 4860 value is 1440 (one day).
- 4861 **Criticality Level (mandatory):** This field defines the level of criticality of this event. The
 4862 action taken by load control devices for an event can be solely based on this value, or
 4863 combination with other Load Control Event fields supported by this device. For example,
 4864 additional fields such as Average Load Adjustment Percentage, Duty Cycle, Cooling
 4865 Temperature Offset, Heating Temperature Offset, Cooling Temperature Set Point or Heating
 4866 Temperature Set Point can be used in combination with the Criticality level. Criticality levels
 4867 are listed in Table D-3.

Table D-3– Criticality Levels

Criticality Level	Level Description	Participation
0	Reserved	
1	Green	Voluntary
2	1	Voluntary
3	2	Voluntary
4	3	Voluntary
5	4	Voluntary
6	5	Voluntary
7	Emergency	Mandatory
8	Planned Outage	Mandatory
9	Service Disconnect	Mandatory
0xA to 0x0F	Utility Defined	Utility Defined
0x10 to 0xFF	Reserved	

4869

- 4870 The criticality level 0x0 and 0x10 to 0xFF are reserved for future profile changes and not used.
- 4871 “Green” event, level 0x01, may be used to denote that the energy delivered uses an abnormal
4872 amount from non-“green” sources. Participation in this event is voluntary.
- 4873 The criticality levels 0x02 through 0x06 (Levels 1 through 5) indicate progressively
4874 increasing levels of load reduction are being requested by the utility. Participation in these events
4875 is voluntary.
- 4876 The criticality level 0x07 is used to indicate an “Emergency” event. Participation in this event is
4877 mandatory, as defined by the utility. The expected response to this event is termination of all
4878 non-essential energy use, as defined by the utility. Exceptions to participation in this event type
4879 must be managed by the utility.
- 4880 The criticality level 0x08 is used to indicate a “Planned Outage” event. Participation in
4881 this event is mandatory, as defined by the utility. The expected response to this event is
4882 termination of delivery of all non-essential energy, as defined by the utility. Exceptions to
4883 participation in this event type must be managed by the utility.
- 4884 The criticality level 0x09 is used to indicate a “Service Disconnect” event. Participation in this
4885 event is mandatory, as defined by the utility. The expected response to this event is
4886 termination of delivery of all non-essential energy, as defined by the utility. Exceptions to
4887 participation in this event type must be managed by the utility.
- 4888 Levels 0x0A to 0x0F are available for Utility Defined criticality levels.
- 4889 **Cooling Temperature Offset (optional):** Requested offset to apply to the normal cooling
4890 setpoint at the time of the start of the event in + 0.1 °C.
- 4891 **Heating Temperature Offset (optional):** Requested offset to apply to the normal heating
4892 setpoint at the time of the start of the event in + 0.1 °C.
- 4893 The Cooling and Heating Temperature Offsets represent a temperature change (Delta
4894 Temperature) that will be applied to both the associated heating and cooling set points. The
4895 temperature offsets (Delta Temperatures) will be calculated per the Local Temperature in the
4896 Thermostat. The calculated temperature will be interpreted as the number of degrees to be
4897 added to the cooling set point and subtracted from the heating set point. Sequential demand
4898 response events are not cumulative. The Offset shall be applied to the normal setpoint.
- 4899 Each offset represents the temperature offset (Delta Temperature) in degrees Celsius, as
4900 follows: Delta Temperature Offset / 10 = delta temperature in degrees Celsius. Where 0.00°C <=
4901 temperature <= 25.4 °C, corresponding to a Temperature in the range 0x00 to 0x0FE. The
4902 maximum resolution this format allowed is 0.1 °C.

- 4903 A DeltaTemperature of 0xFF indicates that the temperature offset is not used.
- 4904 If a temperature offset is sent that causes the heating or cooling temperature set point to exceed
4905 the limit boundaries that are programmed into the thermostat, the thermostat should respond by
4906 setting the temperature at the limit.
- 4907 **Cooling Temperature Set Point (optional):** Requested cooling set point in 0.01 degrees
4908 Celsius.
- 4909 **Heating Temperature Set Point (optional):** Requested heating set point in 0.01 degrees
4910 Celsius.
- 4911 Cooling and heating temperature set points will be defined and calculated per the
4912 *LocalTemperature* attribute in the Thermostat Cluster [B1].
- 4913 These fields represent the temperature in degrees Celsius, as follows:
- 4914 Cooling Temperature Set Point / 100 = temperature in degrees Celsius
- 4915 Where $-273.15^{\circ}\text{C} \leq \text{temperature} \leq 327.67^{\circ}\text{C}$, corresponding to a Cooling and/or Heating
4916 Temperature Set Point in the range 0x954d to 0x7fff.
- 4917 The maximum resolution this format allows is 0.01°C .
- 4918 A Cooling or Heating Temperature Set Point of 0x8000 indicates that the temperature set
4919 point is not used.
- 4920 If a temperature is sent that exceeds the temperature limit boundaries that are programmed
4921 into the thermostat, the thermostat should respond by setting the temperature at the limit.
- 4922 The thermostat shall not use a Cooling or Heating Temperature Set Point that causes the
4923 device to use more energy than the normal setting.
- 4924 When both a Temperature Offset and a Temperature Set Point are provided, the thermostat may
4925 use either as defined by the device manufacturer. The thermostat should use the setting that
4926 provides the lowest energy consumption.
- 4927 **Average Load Adjustment Percentage (optional):** Defines a maximum energy usage limit as
4928 a percentage of the client implementations specific average energy usage. The load adjustment
4929 percentage is added to 100% creating a percentage limit applied to the client implementations
4930 specific average energy usage. A -10% load adjustment percentage will establish an energy
4931 usage limit equal to 90% of the client implementations specific average energy usage. Each
4932 load adjustment percentage is referenced to the client implementations specific average energy
4933 usage. There are no cumulative effects.

4934 The range of this field is -100 to +100 with a resolution of 1 percent. A -100% value equals a
 4935 total load shed. A 0% value will limit the energy usage to the client implementation's specific
 4936 average energy usage. A +100% value will limit the energy usage to double the client
 4937 implementation's specific average energy usage.

4938 A value of 0x80 indicates the field is not used. All other values are reserved for future use.

4939 **Duty Cycle (optional): Defines** the maximum On state duty cycle as a percentage of time.
 4940 Example, if the value is 80, the device would be in an “on state” for 80% of the time for the
 4941 duration of the event. Range of the value is 0 to 100. A value of 0xFF indicates the field is not
 4942 used. All other values are reserved for future use.

4943 Duty cycle control is a device specific issue and shall be managed by the device manufacturer. It
 4944 is expected that the duty cycle of the device under control will span the shortest practical time
 4945 period in accordance with the nature of the device under control and the intent of the request
 4946 for demand reduction. For typical Device Classes, three minutes⁷ for each 10% of duty cycle is
 4947 recommended. It is expected that the “off state” will precede the “on state”.

4948 **Event Control (mandatory):** Identifies additional control options for the event. The BitMap for
 4949 this field is described in Table D-4.

4950

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

4951

4952 *Note: The randomization attribute will be used in combination with two bits to determine if*
 4953 *the Event Start and Stop Times are randomized. By default devices will randomize the start*
 4954 *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*
 4955 *these values.*

4956 **D.2.2.3.1.1.2 When Generated**

4957 This command is generated when the ESI wants to control one or more load control devices,
 4958 usually as the result of an energy curtailment command from the Smart Energy network.

4959 **D.2.2.3.1.1.3 Responses to Load Control Event**

4960 The server receives the cluster-specific commands detailed in sub-clause D.2.3.3.1.

4961 D.2.2.3.2 Cancel Load Control Event Command

4962 D.2.2.3.2.1 Payload Format

4963 The *Cancel Load Control Event* command payload shall be formatted as illustrated in
4964 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)

4965 **Figure D-3—Cancel Load Control Event Payload**

4966 D.2.2.3.2.1.1 *Payload Details*

4967 **Issuer Event ID (mandatory):** Unique identifier generated by the Energy provider. The value of
4968 this field allows matching of Event reports with a specific Demand Response and Load Control
4969 event. It's expected the value contained in this field is a unique number managed by upstream
4970 systems or a UTC based time stamp (UTCTime data type) identifying when the Load Control
4971 Event was issued.

4972 **Device Class (mandatory):** Bit encoded field representing the Device Class to apply the
4973 current Load Control Event. Each bit, if set individually or in combination, indicates the class
4974 device(s) needing to participate in the event. (Note that the participating device may be
4975 different than the controlling device. For instance, a thermostat may act on behalf of an HVAC
4976 compressor or furnace and/or Strip Heat/Baseboard Heater and should take action on their
4977 behalf, as the thermostat itself is not subject to load shed but controls devices that are subject to
4978 load shed.) The encoding of the Device Class is listed in Figure D-2.

4979 **Utility Enrollment Group (mandatory):** The Utility Enrollment Group field can be used in
4980 conjunction with the Device Class bits. It provides a mechanism to direct Load Control Events
4981 to groups of Devices. Example, by assigning two different groups relating to either Demand
4982 Response programs or geographic areas, Load Control Events can be further directed for a sub-
4983 set of Device Classes (i.e. Device Class Bit 0 and Utility Enrollment Group #1 vs. Device
4984 Class Bit0 and Utility Enrollment Group #2). 0x00 addresses all groups, and values 0x01 to
4985 0xFF address individual groups that match. Please refer to sub-clause D.2.3.2.1 for further
4986 details.

4987 If the Device Class and/or Utility Enrollment Group fields don't apply to your End Device, the
4988 *Cancel Load Control Event* command is ignored.

4989 Device Class and/or Utility Group fields must be the same for a *Cancel Load Control Event*
4990 command as they were for the command to create the event. Should these fields be different

4991 there is no defined behavior for how DRLC servers should maintain their tables for replying to
 4992 *Get Scheduled Events* commands.

4993 **Cancel Control (mandatory):** The encoding of the Cancel Control is listed in Table D-5.

4994 **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

4995
 4996 **Effective Time (mandatory):** UTC Timestamp representing when the canceling of the event is
 4997 scheduled to start. An effective time of 0x00000000 is a special time denoting “now.” If the
 4998 device would send an event with an Effective Time of now, adjust the Duration In Minutes field
 4999 to correspond to the remainder of the event.

5000 **Note:** This field is deprecated; a *Cancel Load Control* command shall now take immediate
 5001 effect. A value of 0x00000000 shall be used in all *Cancel Load Control* commands

5002 **D.2.2.3.2.1.2 When Generated**

5003 This command is generated when the ESI wants to cancel previously scheduled control of one or
 5004 more load control devices, usually as the result of an energy curtailment command from the
 5005 Smart Energy network.

5006 **D.2.2.3.2.1.3 Responses to Cancel Load Control Event**

5007 The server receives the cluster-specific commands detailed in sub-clause D.2.3.3.1.

5008 **Note:** If the *Cancel Load Control Event* command is received after the event has ended, the
 5009 device shall reply using the “Report Event Status Command” with an Event Status of “Rejected
 5010 - Invalid Cancel Command (Undefined Event)”.

5011 **D.2.2.3.3 Cancel All Load Control Events Command**

5012 **D.2.2.3.3.1 Payload Format**

5013 The *Cancel All Load Control Events* command payload shall be formatted as illustrated in
 5014 Figure D-4.

5015

Octets	1
Data Type	8-bit BitMap
Field Name	Cancel Control

5016

Figure D-4– Cancel All Load Control Events Command Payload5017 **D.2.2.3.3.1.1 Payload Details**5018 **Cancel Control:** The encoding of the Cancel Control is listed in Table D-6.5019 **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

5020

5021 **D.2.2.3.3.2 When Generated**

5022 This command is generated when the ESI wants to cancel all events for control device(s).

5023 **D.2.2.3.3 Responses to Cancel All Load Control Events**5024 The server receives the cluster-specific commands detailed in sub-clause D.2.3.3.1. The *Cancel*
5025 *All Load Control Events* command is processed by the device as if individual *Cancel Load*
5026 *Control Event* commands were received for all of the currently stored events in the device. The
5027 device will respond with a “Report Event Status Command” for each individual load control
5028 event canceled.5029 **D.2.2.4 Commands Received**

5030 The server receives the cluster-specific commands detailed in sub-clause D.2.3.3.

5031 **D.2.3 Client**

5032 This section identifies the attributes and commands provided by Client devices.

5033 **D.2.3.1 Dependencies**5034 Devices receiving and acting upon *Load Control Event* commands must be capable of storing
5035 and supporting at least three unique instances of events. As a highly recommended recovery
5036 mechanism, when maximum storage of events has been reached and additional Load Control
5037 Events are received that are unique (not superseding currently stored events), devices should

- 5038 ignore additional Load Control Events and when storage becomes available, utilize the
 5039 *GetScheduledEvents* command to retrieve any previously ignored events.
- 5040 Events carried using this cluster include a timestamp with the assumption that target devices
 5041 maintain a real time clock. Devices can acquire and synchronize their internal clocks with the
 5042 ESI as described in sub-clause 5.12.1.1.
- 5043 Devices MAY ‘drop’ events received before they have received and resolved time (‘dropping’ an
 5044 event is defined as sending a default response with status code SUCCESS).
- 5045 If a device does not support a real time clock, it’s assumed the device will ignore all values
 5046 within the Time field except the “Start Now” value.
- 5047 Additionally, for devices without a real time clock it’s assumed those devices will utilize a
 5048 method (i.e. ticks, countdowns, etc.) to approximate the correct duration period.

5049 D.2.3.2 Client Cluster Attributes

5050 **Table D-7– Demand Response Client Cluster Attributes**

Identifier	Name	Type	Range	Access	Default	Mandatory/ Optional
0x0000	<i>UtilityEnrollmentGroup</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					

- 5051
- 5052 **D.2.3.2.1 Utility Enrollment Group Attribute**
- 5053 The *UtilityEnrollmentGroup* provides a method for utilities to assign devices to groups. In
 5054 other words, Utility defined groups provide a mechanism to arbitrarily group together different
 5055 sets of load control or demand response devices for use as part of a larger utility program. The
 5056 definition of the groups, implied usage, and their assigned values are dictated by the Utilities
 5057 and subsequently used at their discretion, therefore outside the scope of this specification. The
 5058 valid range for this attribute is 0x00 to 0xFF, where 0x00 (the default value) indicates the device
 5059 is a member of all groups and values 0x01 to 0xFF indicates that the device is member of that
 5060 specified group.

5061 D.2.3.2.2 **Start Randomization Minutes Attribute**

5062 The *StartRandomizedMinutes* represents the maximum number of minutes to be used when
5063 randomizing the start of an event. As an example, if *StartRandomizedMinutes* is set for 3
5064 minutes, the device could randomly select 2 minutes (but never greater than the 3 minutes) for
5065 this event, causing the start of the event to be delayed by two minutes. The valid range for this
5066 attribute is 0x00 to 0x3C where 0x00 indicates start event randomization is not performed.

5067 D.2.3.2.3 **End Randomization Minutes Attribute**

5068 The *EndRandomizedMinutes* represents the maximum number of minutes to be used when
5069 randomizing the end of an event. As an example, if *EndRandomizedMinutes* is set for 3 minutes,
5070 the device could randomly select one minute (but never greater than 3 minutes) for this event,
5071 causing the end of the event to be delayed by one minute. The valid range for this attribute is
5072 0x00 to 0x3C where 0x00 indicates end event randomization is not performed.

5073 D.2.3.2.4 **DeviceClassValue Attribute**

5074 The *DeviceClassValue* attribute identifies which bits the device will match in the Device Class
5075 fields. Please refer to Table D-2, “Device Class Field BitMap/ Encoding” for further details.
5076 Although the attribute has a read/write access property, the device is permitted to refuse to
5077 change the *DeviceClass* by setting the status field of the corresponding write attribute
5078 status record to NOT_AUTHORIZED.

5079 Although, for backwards compatibility, the Type cannot be changed, this 16-bit Integer should be
5080 treated as if it were a 16-bit BitMap.

5081 Device Class and/or Utility Enrollment Group fields are to be used as filters for deciding to
5082 accept or ignore a *Load Control Event* or a *Cancel Load Control Event* command. There is no
5083 requirement for a device to store or remember the Device Class and/or Utility Enrollment Group
5084 once the decision to accept the event has been made. A consequence of this is that devices that
5085 accept multiple device classes may have an event created for one device class superseded by an
5086 event created for another device class.

5087 In-Home Displays should report the device classes that they are interested in. An IHD that
5088 wishes to display all possible Load Control Events, even for classes not yet defined, should
5089 indicate a device class of 0xFFFF; this will allow DRLC servers to optimize the number of
5090 DRLC events they unicast, such that they are only sent to those devices that are interested in
5091 them.

5092 D.2.3.3 Commands Generated

5093 The command IDs generated by the Demand Response and Load Control client cluster are
5094 listed in Table D-8.

5095

5096

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	

5097

D.2.3.3.1 Report Event Status Command**D.2.3.3.1.1 Payload Format**

5100 The *Report Event Status* command payload shall be formatted as illustrated in Figure D-5.

5101

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)

5102

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)

5103

Figure D-5– Report Event Status Command Payload**D.2.3.3.1.1.1 Payload Details**

5105 **Issuer Event ID (mandatory):** Unique identifier generated by the Energy provider. The value of this field allows matching of Event reports with a specific Demand Response and Load Control event. It's expected the value contained in this field is a unique number managed by upstream systems or a UTC based time stamp (UTCTime data type) identifying when the Load Control Event was issued.

5110 **Event Status (mandatory):** Table D-9 lists the valid values returned in the Event Status field.

5111

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.

5112

5113 Should a device issue one or more “OptOut” or “OptIn” RES commands during an event that
 5114 is eventually cancelled, the event shall be recorded as a cancelled event (Status = 0x06) at its
 5115 effective time.

5116 Should a device issue one or more “OptOut” or “OptIn” RES commands during an event that
 5117 is not cancelled, the event shall be recorded as partially completed based on the last RES
 5118 command sent (Status = 0x08 or 0x09).

5119 When a device returns a status of 0xFD (Rejected - Invalid Cancel Command (Undefined
 5120 Event)), all optional fields should report their “Ignore” values.

5121 When a device receives a duplicate RES command, it should ignore the duplicate commands.
 5122 Please note: As a recommended best practice, ESI applications should provide a mechanism to
 5123 assist in filtering duplicate messages received on the WAN.

- 5124 **Event Status Time (mandatory):** UTC Timestamp representing when the event status
 5125 occurred. This field shall not use the value of 0x00000000.
- 5126 **Criticality Level Applied (mandatory):** Criticality Level value applied by the device, see the
 5127 corresponding field in the *Load Control Event* command for more information.
- 5128 **Cooling Temperature Set Point Applied (optional):** Cooling Temperature Set Point value
 5129 applied by the device, see the corresponding field in the *Load Control Event* command for more
 5130 information. The value 0x8000 means that this field has not been used by the end device.
- 5131 **Heating Temperature Set Point Applied (optional):** Heating Temperature Set Point value
 5132 applied by the device, see the corresponding field in the *Load Control Event* command for more
 5133 information. The value 0x8000 means that this field has not been used by the end device.
- 5134 **Average Load Adjustment Percentage Applied (optional):** Average Load Adjustment
 5135 Percentage value applied by the device, see the corresponding field in the *Load Control Event*
 5136 command for more information. The value 0x80 means that this field has not been used by the
 5137 end device.
- 5138 **Duty Cycle Applied (optional):** Defines the maximum On state duty cycle applied by the
 5139 device. The value 0xFF means that this field has not been used by the end device. Refer to sub-
 5140 clause D.2.2.3.1.1.
- 5141 **Event Control (mandatory):** Identifies additional control options for the event. Refer to sub-
 5142 clause D.2.2.3.1.1.
- 5143 **Signature Type (mandatory):** An 8-bit Unsigned integer enumerating the type of algorithm
 5144 use to create the Signature. The enumerated values are:

Enumerated Value	Signature Type
0x00	No Signature
0x01	ECDSA
0x02 to 0xFF	Reserved

- 5145
 5146 If the signature field is not used, the signature type shall be set to 0x00, which will be used to
 5147 indicate “no signature”. The signature field shall be filled with (48) 0xFF values.
- 5148 **Signature (optional):** A non-repudiation signature created by using the Matyas-Meyer-Oseas
 5149 hash function (specified in Annex B.6 in [B3]) used in conjunction with ECDSA. The
 5150 signature creation process will occur in two steps:

5151 1 Pass the first ten fields, which includes all fields up to the Signature field, of the *Report Event*
 5152 *Status* command (listed in Figure D-5) through ECDSA using the device's ECC Private Key,
 5153 generating the signature (r,s) .

5154 *Note: ECDSA internally uses the MMO hash function in place of the internal SHA-1 hash*
 5155 *function.*

5156 2 Concatenate ECDSA signature components (r,s) and place into the Signature field within the
 5157 *Report Event Status* command.

5158 *Note: the lengths of r and s are implicit, based on the curve used. Verifying the signature will require*
 5159 *breaking the signature field back into the discrete components r and s, based on the length.*

5160 D.2.3.3.1.2 When Generated

5161 This command is generated when the client device detects a change of state for an active Load
 5162 Control event. (The transmission of this command should be delayed after a random delay
 5163 between 0 and 5 seconds, to avoid a potential storm of packets.)

5164 D.2.3.3.2 Get Scheduled Events Command

5165 *Note: The handling of this command is currently under review, and is likely to change in the next*
 5166 *revision of the specification. Refer to CCB 1297 (and associated document 12-0180-00) for*
 5167 *further information*

5168 This command is used to request that all scheduled Load Control Events, starting at or after the
 5169 supplied Start Time, are re-issued to the requesting device. When received by the Server, one or more
 5170 *Load Control Event* commands (see sub-clause D.2.2.3.1) will be sent covering both active and
 5171 scheduled Load Control Events.

5172 D.2.3.3.2.1 Payload Format

5173 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)

5174 **Figure D-6– Get Scheduled Events Command Payload**

5175 **Start Time (mandatory):** UTC Timestamp representing the minimum ending time for any
 5176 scheduled or currently active events to be resent. If either command has a Start Time of
 5177 0x00000000, replace that Start Time with the current time stamp.

5178 **Number of Events (mandatory):** Represents the maximum number of events to be sent. A
 5179 value of 0 would indicate all available events are to be returned. Example: Number of
 5180 Events = 1 would return the first event with an EndTime greater than or equal to the value of

5181 Start Time field in the *Get Scheduled Events* command (EndTime would be StartTime plus
5182 Duration of the event listed in the device's event table).

5183 D.2.3.3.2.2 When Generated

5184 This command is generated when the client device wishes to verify the available Load Control
5185 Events or after a loss of power/reset occurs and the client device needs to recover currently
5186 active or scheduled Load Control Events.

5187 A ZCL Default Response with status NOT_FOUND shall be returned when there are no events
5188 available.

5189 **D.2.3.4 Commands Received**

5190 The client receives the cluster-specific commands detailed in sub-clause D.2.2.

5191 **D.2.3.5 Attribute Reporting**

5192 Attribute reporting is not expected to be used for this cluster. The Client side attributes are
5193 not expected to be changed by the Client, only used during Client operations.

5194 **D.2.4 Application Guidelines**

5195 The criticality level is sent by the utility to the load control device to indicate how much load
5196 reduction is requested. The utility is not required to use all of the criticality levels that are
5197 described in this specification. A load control device is not required to provide a unique
5198 response to each criticality level that it may receive.

5199 The Average Load Adjustment Percentage, temperature offsets, and temperature set points are
5200 used by load control devices and energy management systems on a “voluntary” or “optional”
5201 basis. These devices are not required to use the values that are provided by the utility. They are
5202 provided as a recommendation by the utility.

5203 The load control device shall, in a manner that is consistent with this specification, accurately
5204 report event participation by way of the Report Event Status message.

5205 The Average Load Adjustment Percentage is sent by the utility to the load control device to
5206 indicate how much load reduction is requested. The load control device may respond to this
5207 information in a unique manner as defined by the device manufacturer.

5208 The Duty Cycle is sent by the utility to the load control device to indicate the maximum “On
5209 state” for a device. The control device may respond to this information in a unique manner as
5210 defined by the device manufacturer.

5211 The cooling temperature offset may be sent by the utility to the load shed control to indicate how
5212 much indoor cooling temperature offset is requested. Response of a load control device to this

5213 information is not mandatory. The control device may respond to this information in a
5214 unique manner as defined by the device manufacturer.

5215 The heating temperature offset may be sent by the utility to the load control device to indicate how
5216 much indoor heating temperature offset is requested. The control device may respond to this
5217 information in a unique manner as defined by the device manufacturer.

5218 The cooling temperature may be sent by the utility to the load control device to indicate the
5219 indoor cooling temperature setting that is requested. The control device may respond to this
5220 information in a unique manner as defined by the device manufacturer.

5221 The heating temperature may be sent by the utility to the load control device to indicate the
5222 indoor heating temperature setting that is requested. The control device may respond to this
5223 information in a unique manner as defined by the device manufacturer.

5224 *Note: The most recent Load Control Event supersedes any previous Load Control Event
5225 command for the set of Device Classes and groups for a given time. Nested events and
5226 overlapping events are not allowed. The current active event will be terminated if a new event
5227 is started.*

5228 [D.2.4.1 Load Control Rules, Server](#)

5229 [**D.2.4.1.1 Load Control Server, Identifying Use of SetPoint and Offset Fields**](#)

5230 The use of the fields, Heating and Cooling Temperature Set Points and Heating and Cooling
5231 Temperature Offsets is optional. All fields in the payload must be populated. Non-use of
5232 these fields by the Server is indicated by using the following values: 0x8000 for Set Points
5233 and 0xFF for Offsets. When any of these four fields are indicated as optional, they shall be
5234 ignored by the client.

5235 [**D.2.4.1.2 Load Control Server, Editing of Scheduled Events**](#)

5236 Editing of a scheduled demand response event is not allowed. Editing of an active demand
5237 response event is not allowed. Nested events and overlapping events are not allowed. The
5238 current active event will be terminated if a new event is started.

5239 [D.2.4.2 Load Control Rules, Client](#)

5240 [**D.2.4.2.1 Start and Stop Randomization**](#)

5241 When shedding loads (turning a load control device off), the load control device will optionally
5242 apply start time randomization based on the values specified in the Event Control Bits and the
5243 Client's *Start Randomization Minutes* attribute. By default, devices will apply a random delay
5244 as specified by the default values of start and end randomization in the Demand Response
5245 Client Cluster Attributes table.

5246 When ending a load control event, the load control device will support the same randomization
5247 features as provided in the start load control event.

5248 D.2.4.2.2 **Editing of DR Control Parameters**

5249 In Load Control Device and energy management systems, editing of the demand response
5250 control parameters while participating in an active demand response event is not allowed.

5251 D.2.4.2.3 **Response to Price Events + Load Control Events**

5252 The residential system's response to price driven events will be considered in addition to the
5253 residential system's response to demand response events. Demand response events which require
5254 that the residential system is turned off have priority over price driven events. Demand response
5255 events which require that the residential system go to a fixed setting point have priority over price
5256 driven events. In this case, the thermostat shall not use a Cooling or Heating Temperature Set
5257 Point that causes the device to use more energy than the price driven event setting.

5258 D.2.4.2.4 **Opt-Out Messages**

5259 An event override message, "opt-out", will be sent by the load control device or energy
5260 management system if the operator chooses not to participate in a demand response event by
5261 taking action to override the programmed demand reduction response. The override message
5262 will be sent at the start of the event. In the case where the event has been acknowledged and
5263 started, the override message will be sent when the override occurs.

5264 D.2.4.2.5 **Thermostat/HVAC Controls**

5265 A residential HVAC system will be allowed to change mode, from off to Heat, off to Cool, Cool
5266 to Heat, or Heat to Cool, during a voluntary event which is currently active. The HVAC control
5267 must acknowledge the event, as if it was operating, in that mode, at the start of the event. The
5268 HVAC control must obey the event rules that would have been enforced if the system had been
5269 operating in that mode at the start of the active event.

5270 An event override message, "opt-out", will be sent by the load control device or energy
5271 management system if the operator chooses not to participate in a demand response event by
5272 taking action to override the programmed demand reduction response. The override message
5273 will be sent at the start of the event. In the case where the event has been acknowledged and
5274 started, the override message will be sent when the override occurs.

5275 D.2.4.2.6 **Demand Response and Load Control Transaction Examples**

5276 The following example in Figure D-7 depicts the transactions that would take place for two events,
5277 one that is successful and another that is overridden by the user.

5278

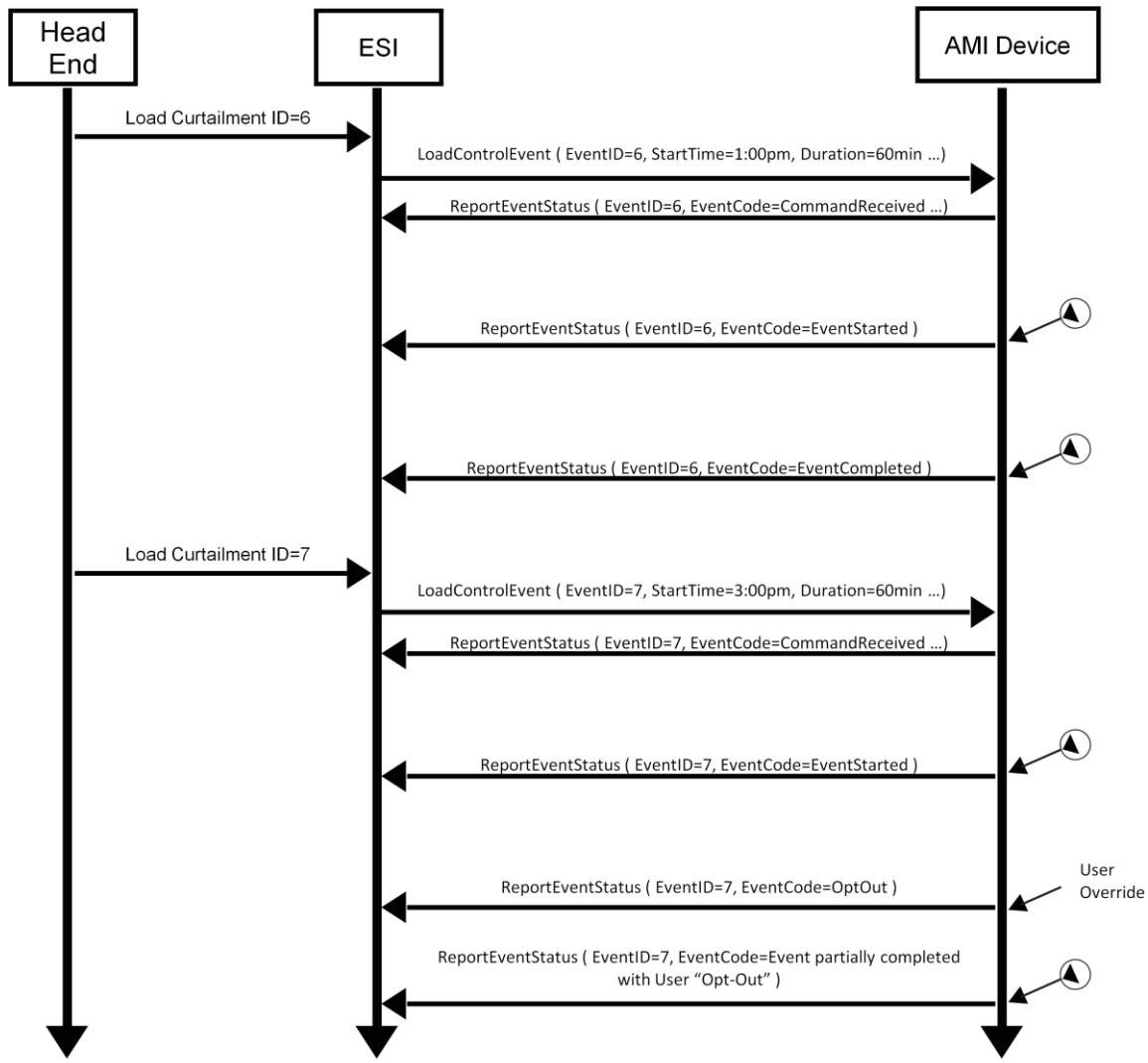
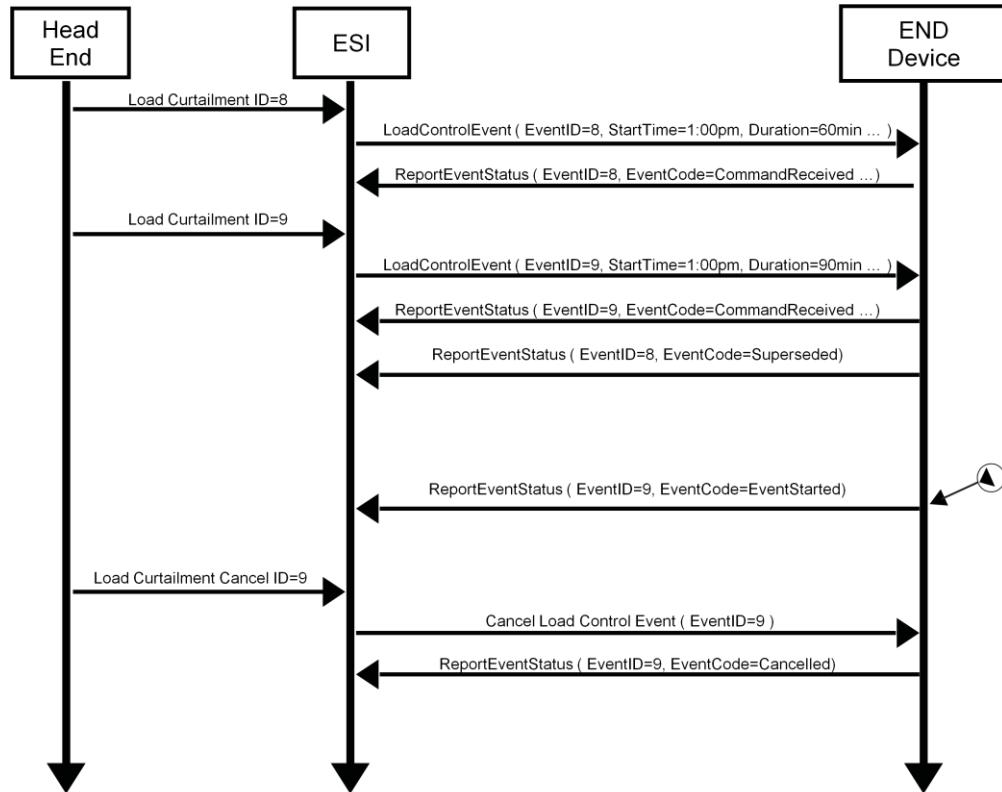
5279
52805281
5282
5283

Figure D-7– Example of Both a Successful and an Overridden Load Curtailment Event

5284 The example in Figure D-8 depicts the transactions that would take place when an event is
 5285 superseded by an event that is eventually cancelled.



5286

5287 **Figure D-8—Example of a Load Curtailment Superseded and**
 5288 **Another Cancelled**

5289

5290 Please refer to Annex E for more information regarding the management and behavior of
 5291 overlapping events.

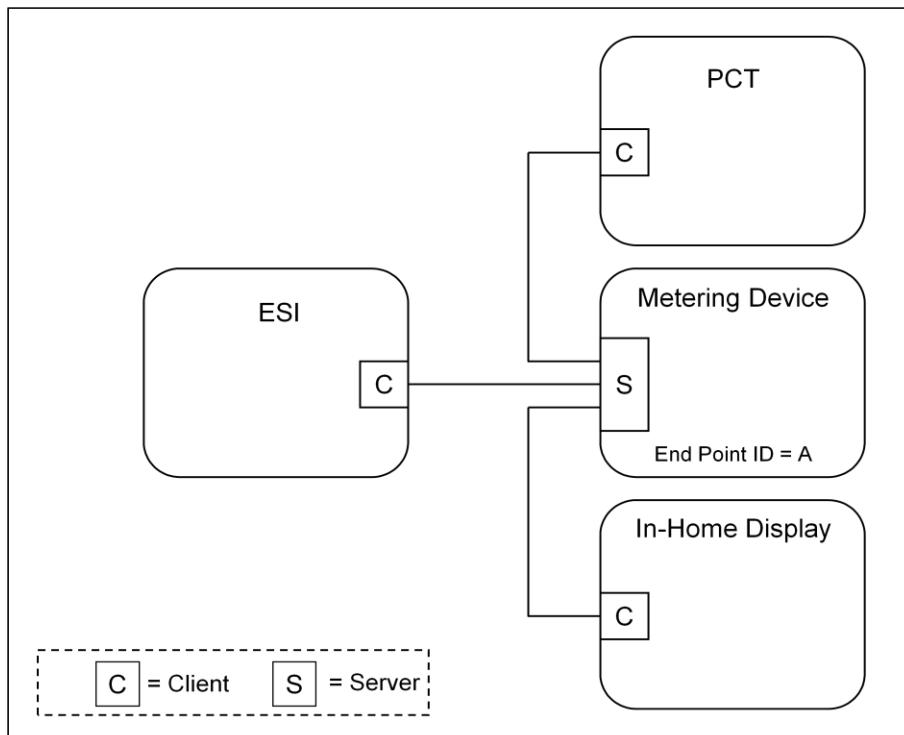
5292

5293

5294 **D.3 Metering Cluster**5295 **D.3.1 Overview**

5296 The Metering Cluster provides a mechanism to retrieve usage information from Electric, Gas,
5297 Water, and potentially Thermal metering devices. These devices can operate on either battery or
5298 mains power, and can have a wide variety of sophistication. The Metering Cluster is designed to
5299 provide flexibility while limiting capabilities to a set number of metered information types. More
5300 advanced forms or data sets from metering devices will be supported in the Smart Energy
5301 Tunneling Cluster, which will be defined in sub-clause D.6.

5302 The following figures identify three configurations as examples utilizing the Metering
5303 Cluster.



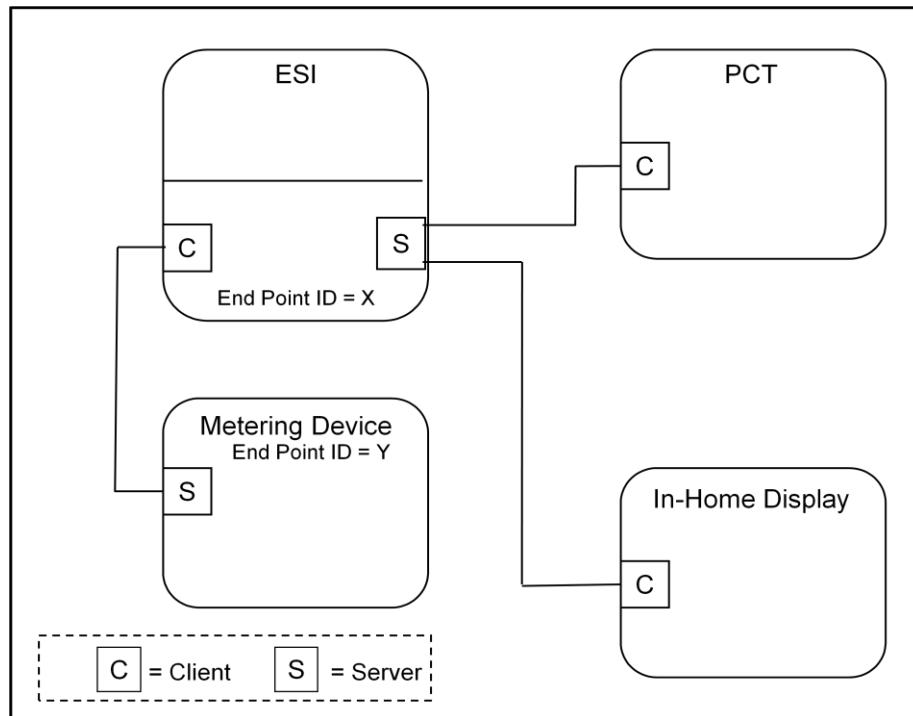
5304
5305 **Figure D-9– Standalone ESI Model with Mains Powered Metering Device**

5306

5307 In the example shown in Figure D-9, the metering device is the source of information
5308 provided via the Metering Cluster Server.

5309

5310



5311
5312

Figure D-10– Standalone ESI Model with Battery Powered Metering Device

5313
5314
5315
5316
5317

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).

5318
5319
5320
5321
5322
5323

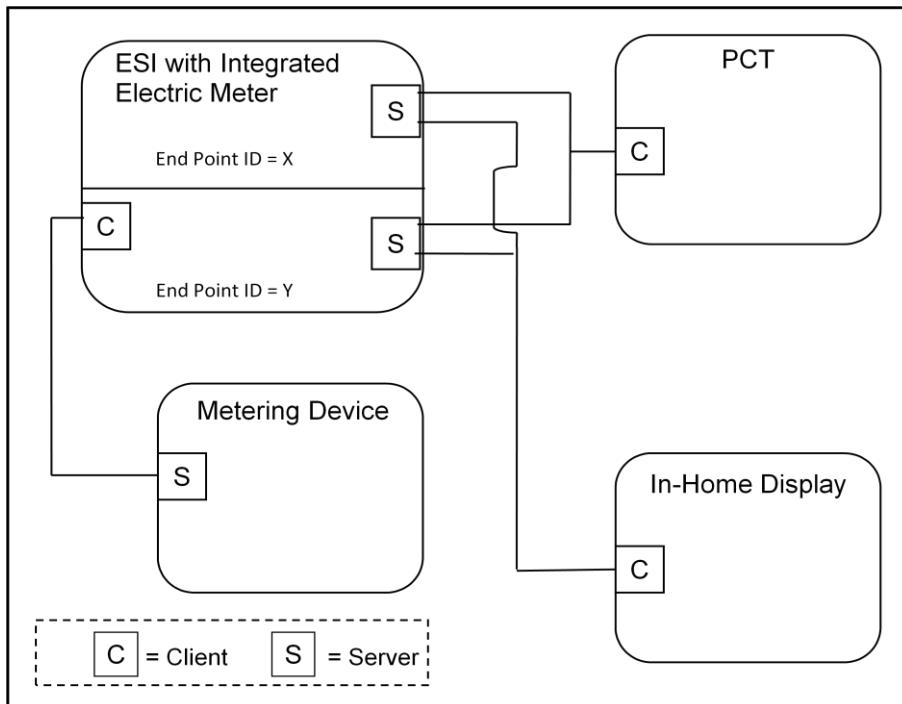
5324
5325

Figure D-11 – ESI Model with Integrated Metering Device

5326
5327 In the example shown in Figure D-11, much like the previous example in Figure D-10,
5328 the external metering device is running on battery power and its duty cycle for providing
5329 information is unknown. It's expected the ESI will act like a Client side mailbox for the external
5330 metering device data, allowing other Smart Energy devices to gain access to the metering
5331 device's data (provided via an image of its Metering Cluster). Since the ESI can also contain an
5332 integrated metering device where its information is also conveyed through the Metering Cluster,
5333 each device (external metering device mailbox and integrated meter) will be available via
5334 independent EndPoint IDs. Other Smart Energy devices that need to access the information must
5335 understand the ESI cluster support by performing service discoveries. It can also identify if an
5336 Endpoint ID is a mailbox/ mirror of a metering device by reading the *MeteringDeviceType*
5337 attribute (refer to sub-clause D.3.2.2.4.7).

5338 In the above examples (Figure D-10 and Figure D-11), it's expected the ESI would perform
5339 Attribute Reads (or configure Attribute Reporting) and use the *GetProfile* command to
5340 receive the latest information whenever the Metering Device (EndPoint Z) wakes up. When
5341 received, the ESI will update its mailbox (EndPoint ID Y in Figure D-10 and Figure D-11) to
5342 reflect the latest data available. A metering device using the mirror is also allowed (and
5343 recommended) to push metering data updates to the ESI via *Report Attribute* commands as
5344 described in sub-clause D.3.4.4.

5345 Other Smart Energy devices can access EndPoint Y in the ESI to receive the latest information
5346 just as they would to access information in the ESI's integrated Electric meter (as in
5347 Figure D-11, EndPoint X) and other Metering devices (as in Figure D-9, EndPoint A).

5348 **D.3.2 Server**

5349 **D.3.2.1 Dependencies**

5350 Subscribed reporting of Metering attributes.

5351 **D.3.2.2 Attributes**

5352 For convenience, the attributes defined in this specification are arranged into sets of related
 5353 attributes; each set can contain up to 256 attributes. Attribute identifiers are encoded such that the
 5354 most significant Octet specifies the attribute set and the least significant Octet specifies the
 5355 attribute within the set. The currently defined attribute sets are listed in Table D-10.

5356 *Note: Certain attributes within this cluster are provisional and not certifiable. Refer to the*
 5357 *individual attribute sets for details of the relevant attributes.*

5358

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

5359

5360 **D.3.2.2.1 Reading Information Set**

5361 The following set of attributes provides a remote access to the reading of the Electric, Gas, or
 5362 Water metering device. A reading must support at least one register which is the actual total
 5363 summation of the delivered quantity (kWh, m³, ft³, ccf, US gl).

5364 Please note: In the following attributes, the term “Delivered” refers to the quantity of Energy,
 5365 Gas, or Water that was delivered to the customer from the utility. Likewise, the term

5366 “Received” refers to the quantity of Energy, Gas, or Water that was received by the utility from
 5367 the customer.

5368 **Note:** *Metering Cluster Reading Attributes 0x0010-0x0014 in this revision of this*
 5369 *specification are provisional and not certifiable. This feature set may change before reaching*
 5370 *certifiable status in a future revision of this specification.*

5371

Table D-11– Reading Information Attribute Set

Identifier	Name	Type	Range	Access	Default	Man./Opt.
0x0000	<i>CurrentSummationDelivered</i>	Unsigned 48-bit Integer	0x00000000000000 to 0xFFFFFFFFFFFFFF	Read Only	-	M
0x0001	<i>CurrentSummationReceived</i>	Unsigned 48-bit Integer	0x00000000000000 to 0xFFFFFFFFFFFFFF	Read Only	-	O
0x0002	<i>CurrentMaxDemandDelivered</i>	Unsigned 48-bit Integer	0x00000000000000 to 0xFFFFFFFFFFFFFF	Read Only	-	O
0x0003	<i>CurrentMaxDemandReceived</i>	Unsigned 48-bit Integer	0x00000000000000 to 0xFFFFFFFFFFFFFF	Read Only	-	O
0x0004	<i>DFTSummation</i>	Unsigned 48-bit Integer	0x00000000000000 to 0xFFFFFFFFFFFFFF	Read Only	-	O
0x0005	<i>Daily Freeze Time</i>	Unsigned 16-bit Integer	0x0000 to 0x183B	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	0x00000000000000 to 0xFFFFFFFFFFFFFF	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>IntervalRead ReportingPeriod</i>	Unsigned 16-bit Integer	0x0000 to 0xFFFF	Read Only	0	O
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	0x000000000000 to 0xFFFFFFFFFFFFFF	Read Only	-	M:Heat M:Cooling O:others
0x0016	<i>CurrentOutletEnergy Carrier Summation</i>	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFFFF	Read Only	-	O
0x0017	<i>InletTemperature</i>	Signed 24-bit Integer	-8,388,607 to 8,388,607	Read Only	-	M:Heat M:Cooling O:others
0x0018	<i>OutletTemperature</i>	Signed 24-bit Integer	-8,388,607 to 8,388,607	Read Only	-	M:Heat M:Cooling O:others
0x0019	<i>ControlTemperature</i>	Signed 24-bit Integer	-8,388,607 to 8,388,607	Read Only	-	O
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	0x00000000000000 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					

5372

5373 D.3.2.2.1.1 CurrentSummationDelivered Attribute

5374 *CurrentSummationDelivered* represents the most recent summed value of Energy, Gas, or Water
 5375 delivered and consumed in the premises. *CurrentSummationDelivered* is mandatory and must be
 5376 provided as part of the minimum data set to be provided by the metering device.
 5377 *CurrentSummationDelivered* is updated continuously as new measurements are made.

5378 D.3.2.2.1.2 CurrentSummationReceived Attribute

5379 *CurrentSummationReceived* represents the most recent summed value of Energy, Gas, or Water
 5380 generated and delivered from the premises. If optionally provided, *CurrentSummationReceived* is
 5381 updated continuously as new measurements are made.

5382 D.3.2.2.1.3 CurrentMaxDemandDelivered Attribute

5383 *CurrentMaxDemandDelivered* represents the maximum demand or rate of delivered
 5384 value of Energy, Gas, or Water being utilized at the premises. If optionally provided,
 5385 *CurrentMaxDemandDelivered* is updated continuously as new measurements are made.

5386 D.3.2.2.1.4 CurrentMaxDemandReceived Attribute

5387 *CurrentMaxDemandReceived* represents the maximum demand or rate of received value of
 5388 Energy, Gas, or Water being utilized by the utility. If optionally provided,
 5389 *CurrentMaxDemandReceived* is updated continuously as new measurements are made.

5390 D.3.2.2.1.5 DFTSummation Attribute

5391 *DFTSummation* represents a snapshot of attribute *CurrentSummationDelivered* captured at the
5392 time indicated by attribute *DailyFreezeTime*. If optionally provided, *DFTSummation* is updated
5393 once every 24 hours and captured at the time set in sub-clause D.3.2.2.1.6.

5394 D.3.2.2.1.6 DailyFreezeTime Attribute

5395 *DailyFreezeTime* represents the time of day when *DFTSummation* is captured. *DailyFreezeTime*
5396 is an unsigned 16-bit value representing the hour and minutes for DFT. The byte usages are:

5397 **Bits 0 to 7:** Range of 0 to 0x3B representing the number of minutes past the top of the hour.

5398

5399 **Bits 8 to 15:** Range of 0 to 0x18 representing the hour of the day (in 24-hour format). *Note*
5400 *that the value 0x18 is only for use in the time 24:00:00.*

5401 D.3.2.2.1.7 PowerFactor Attribute

5402 *PowerFactor* contains the Average Power Factor ratio in 1/100ths. Valid values are 0 to 99.

5403 D.3.2.2.1.8 ReadingSnapshotTime Attribute

5404 The *ReadingSnapshotTime* attribute represents the last time all of the
5405 *CurrentSummationDelivered*, *CurrentSummationReceived*, *CurrentMaxDemandDelivered*, and
5406 *CurrentMaxDemandReceived* attributes that are supported by the device were updated.

5407 D.3.2.2.1.9 CurrentMaxDemandDeliveredTime Attribute

5408 The *CurrentMaxDemandDeliveredTime* attribute represents the time when
5409 *CurrentMaxDemandDelivered* reading was captured.

5410 D.3.2.2.1.10 CurrentMaxDemandReceivedTime Attribute

5411 The *CurrentMaxDemandReceivedTime* attribute represents the time when
5412 *CurrentMaxDemandReceived* reading was captured.

5413 D.3.2.2.1.11 DefaultUpdatePeriod Attribute

5414 The *DefaultUpdatePeriod* attribute represents the interval (seconds) at which the
5415 *InstantaneousDemand* attribute is updated when not in fast poll mode. *InstantaneousDemand*
5416 may be continuously updated as new measurements are acquired, but at a minimum
5417 *InstantaneousDemand* must be updated at the *DefaultUpdatePeriod*. The *DefaultUpdatePeriod*
5418 may apply to other attributes as defined by the device manufacturer.

5419 D.3.2.2.1.12 FastPollUpdatePeriod Attribute

5420 The *FastPollUpdatePeriod* attribute represents the interval (seconds) at which the
5421 *InstantaneousDemand* attribute is updated when in fast poll mode. *InstantaneousDemand* may be
5422 continuously updated as new measurements are acquired, but at a minimum,

5423 *InstantaneousDemand* must be updated at the *FastPollUpdatePeriod*. The
5424 *FastPollUpdatePeriod* may apply to other attributes as defined by the device manufacturer.

5425 D.3.2.2.1.13 CurrentBlockPeriodConsumptionDelivered Attribute

5426 The *CurrentBlockPeriodConsumptionDelivered* attribute represents the most recent summed
5427 value of Energy, Gas or Water delivered and consumed in the premises during the Block Tariff
5428 Period.

5429 The *CurrentBlockPeriodConsumptionDelivered* is reset at the start of each Block Tariff Period.

5430 D.3.2.2.1.14 DailyConsumptionTarget Attribute

5431 The *DailyConsumptionTarget* attribute is a daily target consumption amount that can be
5432 displayed to the consumer on a HAN device, with the intent that it can be used to compare to
5433 actual daily consumption (e.g. compare to the *CurrentDayConsumptionDelivered*).

5434 This may be sent from the utility to the ESI, or it may be derived. Although intended to be
5435 based on Block Thresholds, it can be used for other targets not related to blocks. The
5436 formatting will be based on the *HistoricalConsumptionFormatting* attribute.

5437 Example: If based on a Block Threshold, the *DailyConsumptionTarget* could be calculated
5438 based on the number of days specified in the Block Tariff Period and a given Block Threshold as
5439 follows: $DailyConsumptionTarget = BlockNThreshold / ((BlockPeriodDuration / 60) / 24)$.
5440 Example: If the target is based on a *Block1Threshold* of 675kWh and where 43200
5441 *BlockThresholdPeriod* is the number of minutes in the billing period (30 days), the
5442 *ConsumptionDailyTarget* would be $675 / ((43200 / 60) / 24) = 22.5$ kWh per day.

5443 D.3.2.2.1.15 CurrentBlock Attribute

When Block Tariffs are enabled, *CurrentBlock* is an 8-bit Enumeration which indicates the currently active block. If blocks are active then the current active block is based on the *CurrentBlockPeriodConsumptionDelivered* and the block thresholds. Block 1 is active when the value of *CurrentBlockPeriodConsumptionDelivered* is less than or equal to the³ *Block1Threshold* value, Block 2 is active when *CurrentBlockPeriodConsumptionDelivered* is greater than *Block1Threshold* value and less than or equal to the⁴ *Block2Threshold* value, and so on. Block 16 is active when the value of *CurrentBlockPeriodConsumptionDelivered* is greater than *Block15Threshold* value.

Table D-12– Block Enumerations

Enumerated Value	Register Block
0x00	No Blocks in use

³ CCB 1679

⁴ CCB 1679

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

5453 D.3.2.2.1.16 ProfileIntervalPeriod Attribute

5454 The *ProfileIntervalPeriod* attribute is currently included in the *Get Profile Response* command
 5455 payload, but does not appear in an attribute set. This represents the duration of each interval.
 5456 *ProfileIntervalPeriod* represents the interval or time frame used to capture metered Energy, Gas, and
 5457 Water consumption for profiling purposes. The enumeration for this field shall match one of the
 5458 *ProfileIntervalPeriod* values defined in sub-clause D.3.2.3.1.1.

5459 D.3.2.2.1.17 IntervalReadReportingPeriod Attribute

5460 The *IntervalReadReportingPeriod* attribute represents how often (in minutes) the water or gas
 5461 meter is to wake up and provide interval data. E.g.: If *IntervalReadReportingPeriod* is set to
 5462 360, then every 6 hours the water or gas meter is to wake up and provide 6 hours of interval
 5463 data in a *Get Profile Response* command. If it is set to 5760 then every 4 days it will wake up and
 5464 provide 4 days of interval data in a *Get Profile Response* command. In some cases data may
 5465 overlap data sent in previous *Get Profile Response* command.

5466 D.3.2.2.1.18 PresetReadingTime

5467 The *PresetReadingTime* attribute represents the time of day (in quarter hour increments) at
 5468 which the meter will wake up and report a register reading even if there has been no
 5469 consumption for the previous 24 hours. *PresetReadingTime* is an unsigned 16-bit value
 5470 representing the hour and minutes. The byte usages are:

5471 **Bits 0 to 7:** Range of 0 to 0x3B representing the number of minutes past the top of the hour.

5472 **Bits 8 to 15:** Range of 0 to 0x17 representing the hour of the day (in 24-hour format).

5473 E.g.: A setting of 0x172D would represent 23:45 hours or 11:45 pm; a setting of 0x071E would
 5474 represent 07:30 hours or 7:30 am. A setting of 0xFFFF indicates this feature is disabled. The use
 5475 of Attribute Reporting Configuration is optional.

5476 D.3.2.2.1.19 VolumePerReport Attribute

5477 The *VolumePerReport* attribute represents the volume per report increment from the water or
 5478 gas meter. For example a gas meter might be set to report its register reading for every time 1
 5479 cubic meter of gas is used. For a water meter it might report the register value every 10 liters of
 5480 water usage.

5481 D.3.2.2.1.20 FlowRestriction Attribute

5482 The *FlowRestriction* attribute represents the volume per minute limit set in the flow restrictor.
 5483 This applies to water but not for gas. A setting of 0xFF indicates this feature is disabled.

5484 D.3.2.2.1.21 SupplyStatus Attribute

5485 The *SupplyStatus* attribute represents the state of the supply at the customer's premises. The
 5486 enumerated values for this field are outlined in Table D-13:

5487 **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

5488

5489 D.3.2.2.1.22 CurrentInletEnergyCarrierSummation Attribute

5490 *CurrentInletEnergyCarrierSummation* is the current integrated volume of a given energy carrier
 5491 measured on the inlet. The formatting and unit of measure for this value is specified in the
 5492 *EnergyCarrierUnitOfMeasure* and *EnergyCarrierSummationFormatting* attributes (refer to Table
 5493 D-25).

5494 The Energy consumption registered in *CurrentSummationDelivered* is not necessarily a direct
 5495 function of this value. The quality of the energy carrier may vary from day to day, e.g. Gas may
 5496 have different quality.

5497 For heat and cooling meters the energy carrier is water at high or low temperature, the energy
 5498 withdrawn from such a system is a function of the flow and the inlet and outlet temperature.

5499 D.3.2.2.1.23 CurrentOutletEnergyCarrierSummation Attribute

5500 *CurrentOutletEnergyCarrierSummation* is the current integrated volume of a given energy
5501 carrier measured on the outlet. The formatting and unit of measure for this value is specified in
5502 the *EnergyCarrierUnitOfMeasure* and *EnergyCarrierSummationFormatting* attributes (refer to
5503 Table D-25).

5504 D.3.2.2.1.24 InletTemperature Attribute

5505 *InletTemperature* is the temperature measured on the energy carrier inlet.

5506 The formatting and unit of measure for this value is specified in the *TemperatureUnitOfMeasure*
5507 and *TemperatureFormatting* attributes (refer to Table D-25).

5508 D.3.2.2.1.25 OutletTemperature Attribute

5509 *OutletTemperature* is the temperature measured on the energy carrier outlet.

5510 The formatting and unit of measure for this value is specified in the *TemperatureUnitOfMeasure*
5511 and *TemperatureFormatting* attributes (refer to Table D-25).

5512 D.3.2.2.1.26 ControlTemperature Attribute

5513 *ControlTemperature* is a reference temperature measured on the meter used to validate the
5514 Inlet/Outlet temperatures.

5515 The formatting and unit of measure for this value is specified in the
5516 *TemperatureUnitOfMeasure* and *TemperatureFormatting* attributes (refer to Table D-25).

5517 D.3.2.2.1.27 CurrentInletEnergyCarrierDemand Attribute

5518 *CurrentInletEnergyCarrierDemand* is the current absolute demand on the energy carrier inlet.

5519 The formatting and unit of measure for this value is specified in the
5520 *EnergyCarrierUnitOfMeasure* and *EnergyCarrierDemandFormatting* attributes (refer to Table
5521 D-25).

5522 For a heat or cooling meter this will be the current absolute flow rate measured on the inlet.

5523 D.3.2.2.1.28 CurrentOutletEnergyCarrierDemand Attribute

5524 *CurrentOutletEnergyCarrierDemand* is the current absolute demand on the energy carrier
5525 outlet.

5526 The formatting and unit of measure for this value is specified in the
5527 *EnergyCarrierUnitOfMeasure* and *EnergyCarrierDemandFormatting* attributes (refer to Table
5528 D-25).

5529 For a heat or cooling meter this will be the current absolute flow rate measured on the outlet.

5530 D.3.2.2.1.29 PreviousBlockPeriodConsumptionDelivered Attribute

5531 The *PreviousBlockPeriodConsumptionDelivered* attribute represents the total value of Energy,
5532 Gas or Water delivered and consumed in the premises at the end of the previous Block Tariff
5533 Period. If supported, the *PreviousBlockPeriodConsumptionDelivered* attribute is updated at the
5534 end of each Block Tariff Period.

5535 D.3.2.2.1.30 CurrentBlockPeriodConsumptionReceived Attribute

5536 The *CurrentBlockPeriodConsumptionReceived* attribute represents the most recent summed
5537 value of Energy, Gas or Water received by the energy supplier from the premises during the
5538 Block Tariff Period. The *CurrentBlockPeriodConsumptionReceived* attribute is reset at the start
5539 of each Block Tariff Period.

5540 D.3.2.2.1.31 CurrentBlockReceived Attribute

5541 When Block Tariffs are enabled, *CurrentBlockReceived* is an 8-bit Enumeration which indicates
5542 the currently active block. If blocks are active then the current active block is based on the
5543 *CurrentBlockPeriodConsumptionReceived* and the block thresholds. Block 1 is active when the
5544 value of *CurrentBlockPeriodConsumptionReceived* is less than or equal to the Block1Threshold
5545 value, Block 2 is active when *CurrentBlockPeriodConsumptionReceived* is greater than
5546 Block1Threshold value and less than or equal to the Block2Threshold value, and so on. Block 16
5547 is active when the value of *CurrentBlockPeriodConsumptionReceived* is greater than
5548 Block15Threshold value.

5549 Refer to Table D-12 for block enumerations.

5550 D.3.2.2.1.32 DFTSummationReceived Attribute

5551 *DFTSummationReceived* represents a snapshot of attribute *CurrentSummationReceived* captured
5552 at the time indicated by the *DailyFreezeTime* attribute (see D.3.2.2.1.6).

5553 If optionally provided, *DFTSummationReceived* is updated once every 24 hours and captured at
5554 the time set in the *DailyFreezeTime* attribute (see D.3.2.2.1.6).

5555 D.3.2.2.1.33 ActiveRegisterTierDelivered Attribute

5556 The *ActiveRegisterTierDelivered* attribute indicates the current register tier that the energy
5557 consumed is being accumulated against. Valid values for this attribute are defined in Table D-98.

5558 D.3.2.2.1.34 ActiveRegisterTierReceived Attribute

5559 The *ActiveRegisterTierReceived* attribute indicates the current register tier that the energy
5560 generated is being accumulated against. Valid values for this attribute are defined in Table
5561 D-101.

5562 D.3.2.2.1.35 LastBlockSwitchTime Attribute

5563 This attribute allows other devices to determine the time at which a meter switches from one
5564 block to another.

5565 When Block Tariffs are enabled, the *LastBlockSwitchTime* attribute represents the
5566 timestamp of the last update to the *CurrentBlock* attribute, as a result of the consumption
5567 exceeding a threshold, or the start of a new block period and/or billing period.

5568 If, at the start of a new block period and/or billing period, the value of the *CurrentBlock* attribute
5569 is still set to Block1 (0x01), the *CurrentBlock* attribute value will not change but the
5570 *LastBlockSwitchTime* attribute shall be updated to indicate this change.
5571

5572 D.3.2.2.2Summation TOU Information Set

5573 The following set of attributes provides a remote access to the Electric, Gas, or Water metering
5574 device's Time of Use (TOU) readings.

5575 *Note: TOU Information Attribute Set Attributes 0x010C-0x011D in this revision of this*
5576 *specification are provisional and not certifiable. This feature set may change before reaching*
5577 *certifiable status in a future revision of this specification.*

5578

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	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O

0x01FD	Reserved					
0x01FE	<i>CPP2 Summation Delivered</i>	Unsigned 48-bit integer	0x000000000000 to 0xFFFFFFFFFFFF	Read only	-	O
0x01FF	Reserved					

5579

5580 D.3.2.2.2.1 CurrentTierNSummationDelivered Attributes

5581 Attributes *CurrentTier1SummationDelivered* through *CurrentTierNSummationDelivered* 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 or a real time pricing period. If optionally provided, attributes *CurrentTier1SummationDelivered* through *CurrentTierNSummationDelivered* are updated continuously as new measurements are made.

5587 D.3.2.2.2.2 CurrentTierNSummationReceived Attributes

5588 Attributes *CurrentTier1SummationReceived* through *CurrentTierNSummationReceived* represent the most recent summed value of Energy, Gas, or Water provided by the premises (i.e. received by the utility from the customer) at a specific price tier as defined by a TOU schedule or a real time pricing period. If optionally provided, attributes *CurrentTier1SummationReceived* through *CurrentTierNSummationReceived* are updated continuously as new measurements are made.

5593 D.3.2.2.2.3 CPP1SummationDelivered Attribute

5594 *CPP1SummationDelivered* represents the most recent summed value of Energy, Gas, or Water delivered to the premises (i.e. delivered to the customer from the utility) while Critical Peak Price ‘CPP1’ was being applied. If optionally provided, attribute *CPP1SummationDelivered* is updated continuously as new measurements are made.

5598 D.3.2.2.2.4 CPP2SummationDelivered Attribute

5599 *CPP2SummationDelivered* represents the most recent summed value of Energy, Gas, or Water delivered to the premises (i.e. delivered to the customer from the utility) while Critical Peak Price ‘CPP2’ was being applied. If optionally provided, attribute *CPP2SummationDelivered* is updated continuously as new measurements are made.

5603

5604

5605 D.3.2.2.3 Meter Status Attribute Set

5606 The Meter Status Attribute Set is defined in Table D-15.

5607

5608

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	-	M:Heat M:Cooling O:other s
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					

5609

5610 D.3.2.2.3.1 Status Attribute

5611 The *Status* attribute provides indicators reflecting the current error conditions found by the
 5612 metering device. This attribute is an 8-bit field where when an individual bit is set, an error or
 5613 warning condition exists. The behavior causing the setting or resetting each bit is device
 5614 specific. In other words, the application within the metering device will determine and control
 5615 when these settings are either set or cleared. Depending on the commodity type, the bits of this
 5616 attribute will take on different meaning. Tables D.16, D.17, D.18, and D.19 below show the bit
 5617 mappings for the *Status* attribute for Electricity, Gas, Water and Heating/Cooling
 5618 respectively. A battery-operated meter will report any change in state of the *Status* when it
 5619 wakes up via a ZCL report attributes command. The ESI is expected to make alarms
 5620 available to upstream systems together with consumption data collected from the battery
 5621 operated meter.

5622

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
----------	-------------------------	-------------	---------------	---------------	---------------	-------------	-------------

5623

5624 The definitions of the Electricity *Status* bits are:

5625 **Service Disconnect Open:** Set to true when the service has been disconnected to this premises.

5626 **Leak Detect:** Set to true when a leak has been detected.

5627 **Power Quality:** Set to true if a power quality event has been detected such as a low voltage, high voltage.

5629 **Power Failure:** Set to true during a power outage.

5630 **Tamper Detect:** Set to true if a tamper event has been detected.

5631 **Low Battery:** Set to true when the battery needs maintenance.

5632 **Check Meter:** Set to true when a non fatal problem has been detected on the meter such as a measurement error, memory error, self check error.

5634

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

5635

5636 The definitions of the Gas *Status* bits are:

5637 **Reverse Flow:** Set to true if flow detected in the opposite direction to normal (from consumer to supplier).

5639 **Service Disconnect:** Set to true when the service has been disconnected to this premises. Ex. The valve is in the closed position preventing delivery of gas.

5641 **Leak Detect:** Set to true when a leak has been detected.

5642 **Low Pressure:** Set to true when the pressure at the meter is below the meter's low pressure threshold value.

5644 **Tamper Detect:** Set to true if a tamper event has been detected.

5645 **Low Battery:** Set to true when the battery needs maintenance.

Check Meter: Set to true when a non fatal problem has been detected on the meter such as a measurement error, memory error, or self check error.

Table D-18– Meter Status Attribute (Water)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reverse Flow	Service Disconnect	Leak Detect	Low Pressure	Pipe Empty	Tamper Detect	Low Battery	Check Meter

5649

5650 The definitions of the Water Status bits are:

5651 **Reverse Flow:** Set to true if flow detected in the opposite direction to normal (from consumer
5652 to supplier).

5653 **Service Disconnect:** Set to true when the service has been disconnected to this premises. Ex.
5654 The valve is in the closed position preventing delivery of water.

5655 **Leak Detect:** Set to true when a leak has been detected.

5656 **Low Pressure:** Set to true when the pressure at the meter is below the meter's low pressure
5657 threshold value.

5658 **Pipe Empty:** Set to true when the service pipe at the meter is empty and there is no flow in
5659 either direction.

5660 Tamper Detect: Set to true if a tamper event has been detected.

5661 Low Battery: Set to true when the battery needs maintenance.

Check Meter: Set to true when a non fatal problem has been detected on the meter such as a measurement error, memory error, or self check error.

Table D-19— Meter Status Attribute (Heat and Cooling)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Flow Sensor	Service Disconnect	Leak Detect	Burst Detect	Temperature Sensor	Tamper Detect	Low Battery	Check Meter ^b

5665

5666 The definitions of the Heat and Cooling *Status* bits are:

5667 Flow Sensor: Set to true when an error is detected on a flow sensor at this premises.

Service Disconnect: Set to true when the service has been disconnected to this premises. Ex. The valve is in the closed position preventing delivery of heat or cooling.

- 5670 **Leak Detect:** Set to true when a leak has been detected.
- 5671 **Burst Detect:** Set to true when a burst is detected on pipes at this premises.
- 5672 **Temperature Sensor:** Set to true when an error is detected on a temperature sensor at this premises.
- 5673
- 5674 **Tamper Detect:** Set to true if a tamper event has been detected.
- 5675 **Low Battery:** Set to true when the battery needs maintenance.
- 5676 **Check Meter:** Set to true when a non fatal problem has been detected on the meter such as a measurement error, memory error, or self check error.
- 5677
- 5678 *Note: It is not necessary to set aside Bit 7 as an “Extension Bit” for future expansion. If extra status bits are required an Extended Meter Status attribute may be added to support additional status values.*
- 5679
- 5680
- 5681 D.3.2.2.3.2 RemainingBatteryLife Attribute
- 5682 *RemainingBatteryLife* represents the estimated remaining life of the battery in % of capacity. A setting of 0xFF indicates this feature is disabled. The range 0 - 100 where 100 = 100%, 0xFF = Unknown.
- 5683
- 5684
- 5685 D.3.2.2.3.3 HoursInOperation Attribute
- 5686 *HoursInOperation* is a counter that increments once every hour during operation. This may be used as a check for tampering.
- 5687
- 5688 *Note: For meters that are not electricity meters turning off the meter does not necessarily prevent delivery of energy — but the meter might not be able to measure it.*
- 5689
- 5690 D.3.2.2.3.4 HoursInFault Attribute
- 5691 *HoursInFault* is a counter that increments once every hour when the device is in operation with a fault detected. This may be used as a check for tampering.
- 5692
- 5693 *Note: For meters that are not electricity meters turning off the meter does not necessarily prevent delivery of energy - but the meter might not be able to measure it.*
- 5694
- 5695 D.3.2.2.3.5 ExtendedStatus Attribute
- 5696 The ExtendedStatus attribute reflects the state of items in a meter that the standard Status attribute cannot show. The Extended Status BitMap is split into two groups of flags: general flags and metering type specific flags. Flags are currently defined for electricity and gas meters; flag definitions for other commodities will be added as and when their usage is agreed.
- 5697
- 5698
- 5699

5700 These flags are set and reset by the meter autonomously; they cannot be reset by other devices.
 5701 The mapping is as defined in the tables below. A meter which implements the attribute but does
 5702 not implement a specific flag internally will simply have the corresponding bit always set to 0.

5703
 5704

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

5705 The definitions of the General *Extended Status* bits are:

5706 **Meter Cover Removed:** Set to true when the device detects the meter cover being removed.

5707 **Strong Magnetic Field detected:** Set to true when the device detects presence of a strong
 5708 magnetic field.

5709 **Battery Failure:** Set to true when the device detects that its battery has failed.

5710 **Program Memory Error:** Set to true when the device detects an error within its program (non-volatile) memory.

5712 **RAM Error:** Set to true when the device detects an instance of a Random Access Memory (RAM) error within the device memory.

5714 **NV Memory Error:** Set to true when the device detects an instance of a Non Volatile (NV) memory error within the device memory - this is a fatal meter error that will require the meter replacement.

5717 **Measurement System Error:** Set to true when the device detects an error within its measurement system.

5719 **Watchdog Error:** Set to true when the device has detected an instance of a watchdog reset
 5720 event (following a catastrophic fault within the device).

5721 **Supply Disconnect Failure:** Set to true when the device has detected that the valve has not
 5722 closed as expected (for gas) or the contactor has not opened as expected (for electricity).

5723 **Supply Connect Failure:** Set to true when the device has detected that the valve has not opened
 5724 as expected (for gas) or the contactor has not closed as expected (for electricity).

5725 **Measurement SW Changed/Tampered:** Set to true when the device detects that its
 5726 measurement software has changed.

5727 **Clock Invalid:** Set to true when the device detects that its internal clock is invalid.

5728 **Temperature Exceeded:** Set to true when the metering device's temperature exceeds a
 5729 predefined limit. There are various reasons for temperature rise in metering devices.

5730 **Moisture Detected:** Set to true when a sensor has detected the presence of moisture e.g. moisture
 5731 in a gas line which can cause a drop in gas pressure, or moisture detected in the sealed
 5732 component area within a water meter.

5733

5734 **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

5735

5736 The definitions of the Electricity-Meter-Specific *Extended Status* bits are:

5737 **Terminal Cover Removed:** Set to true when the device detects that its terminal cover has been
 5738 removed.

5739 **Incorrect Polarity:** Set to true when the electricity meter detects incorrect polarity on the
 5740 electricity supply.

5741 **Current with No Voltage:** Set to true when the meter has been tampered with, to disconnect the
 5742 measurement function from the supply. Electricity is still flowing but not being recorded.

5743 **Limit Threshold Exceeded:** Set to true when the electricity meter detects that the load has
 5744 exceeded the load limit threshold.

5745 **Under Voltage:** Set to true when the electricity meter indicates that the voltage measurement
 5746 over the voltage measurement period is lower than the voltage threshold.

5747 **Over Voltage:** Set to true when the electricity meter indicates that the voltage measurement over
 5748 the voltage measurement period is higher than the voltage threshold.

5749

5750 **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

5751
 5752 The definitions of the Gas-Meter-Specific *Extended Status* bits are:

5753 **Battery Cover Removed:** Set to true when the gas meter detects that its battery cover has been
 5754 removed.

5755 **Tilt Tamper:** Set to true when the meter detects a change in its physical properties (i.e. that it is
 5756 being tilted, the tilt sensor has been activated or otherwise tampered with).

5757 **Excess Flow:** Set to true when the gas meter detects excess flow (e.g. when local supply
 5758 restoration is attempted).

5759

5760 D.3.2.2.3.6 *RemainingBatteryLifeinDays* Attribute

5761 *RemainingBatteryLifeInDays* attribute represents the estimated remaining life of the battery in
 5762 days of capacity. The range is 0 – 0xFFFF, where 0xFFFF represents 'Invalid', 'Unused' and
 5763 'Disabled'.

5764 D.3.2.2.3.7 *CurrentMeterID* Attribute

5765 *CurrentMeterID* attribute is the current id for the Meter. This could be the current firmware
 5766 version supported on the meter.

5767 D.3.2.2.3.8 *AmbientConsumptionIndicator* Attribute

5768 The *AmbientConsumptionIndicator* attribute is an 8-bit enumeration which provides a simple (i.e.
 5769 Low/Medium/High) indication of the amount of a commodity being consumed within the

5770 premises. The status is achieved by comparing the current value of the *InstantaneousDemand*
 5771 attribute (see D.3.2.2.5.1) with low/medium and medium/high thresholds. The status is defined in
 5772 Table D-23:

5773 **Table D-23– LowMediumHighStatus Attribute**

Enumeration	Description
0x00	Low Energy usage
0x01	Medium Energy usage
0x02	High Energy usage

5774 The thresholds which are used to determine the value of this attribute are themselves defined as
 5775 attributes within section D.10.2.2.4.1 and section D.10.2.2.4.2

5776

5777 **D.3.2.2.4Formatting**

5778 The following set of attributes provides the ratios and formatting hints required to transform the
 5779 received summations, consumptions, temperatures, or demands/ rates into displayable values.
 5780 If the Multiplier and Divisor attribute values are non-zero, they are used in conjunction with the
 5781 *SummationFormatting*, *ConsumptionFormatting*, *DemandFormatting*, and
 5782 *TemperatureFormatting* attributes.

5783 Equations required to accomplish this task are defined below:

5784 Summation = Summation received * Multiplier / Divisor
 5785 (formatted using *SummationFormatting*)

5786 Consumption = Summation received * Multiplier / Divisor
 5787 (formatted using *ConsumptionFormatting*)

5788 Demand = Demand received * Multiplier / Divisor
 5789 (formatted using *DemandFormatting*)

5790 Temperature = Temperature received * Multiplier / Divisor

5791 If the Multiplier and Divisor attribute values are zero, just the formatting hints defined in
 5792 *SummationFormatting*, *ConsumptionFormatting*, *DemandFormatting* and
 5793 *TemperatureFormatting* attributes are used.

5794 The summation received, consumption received, demand received, and temperature received
 5795 variables used above can be replaced by any of the attributes listed in sub-clauses
 5796 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.

5797 The following table shows examples that demonstrate the relation between these attributes.

5798

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

5799

5800 The Consumption Formatting Attribute Set is defined in Table D-25.

5801 *Note: Consumption Formatting Attribute 0x0307 in this revision of this specification is provisionary and not certifiable. This feature set may change before reaching certifiable status in a future revision of this specification.*
5802
5803

5804

5805

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 0xFFFFFFF	Read Only	-	O
0x0302	<i>Divisor</i>	Unsigned 24-bit Integer	0x000000 0 to 0xFFFFFFF	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>EnergyCarrierUnit OfMeasure</i>	8-bit Enumeration	0x00 to 0xFF	Read Only	-	M:Heat M:Cooling O:others
0x030A	<i>EnergyCarrier SummationFormatting</i>	8-bit BitMap	0x00 to 0xFF	Read Only	-	M:Heat M:Cooling O:others
0x030B	<i>EnergyCarrier DemandFormatting</i>	8-bit BitMap	0x00 to 0xFF	Read Only	-	O
0x030C	<i>TemperatureUnit OfMeasure</i>	8-bit Enumeration	0x00 to 0xFF	Read Only	-	M:Heat M:Cooling O:others
0x030D	<i>TemperatureFormatting</i>	8-bit BitMap	0x00 to 0xFF	Read Only	-	M:Heat M:Cooling O:others
0x030E	<i>ModuleSerialNumber</i>	Octet String	1 to 25 Octets	Read only	-	O
0x030F	<i>OperatingTariffLabel Delivered</i>	Octet String	1 to 25 Octets	Read only	-	O
0x0310	<i>OperatingTariffLabel Received</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>AlternativeUnitof Measure</i>	8-bit Enumeration	0x00 to 0xFF	Read Only	0x00	O
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					

5806

5807 D.3.2.2.4.1 UnitofMeasure Attribute

5808 *UnitofMeasure* provides a label for the Energy, Gas, or Water being measured by the metering
 5809 device. The unit of measure applies to all summations, consumptions/ profile interval and
 5810 demand/rate supported by this cluster other than those specifically identified as being based upon
 5811 the *EnergyCarrierUnitOfMeasure* or the *AlternativeUnitofMeasure*. Other measurements such as
 5812 the power factor are self describing. This attribute is an 8-bit enumerated field. The bit
 5813 descriptions for this Attribute are listed in Table D-26.

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.

5816

5817 **Note:** When using BCD for meter reads, the values A to F are special values or indicators
5818 denoting “Opens”, “Shorts”, and etc. conditions when reading meter register hardware. Any
5819 SE device displaying the BCD based values to end users should use a non-decimal value to
5820 replace the A to F. In other words, a device could use an “*” in place of the special values or
5821 indicators.

5822 D.3.2.2.4.2 Multiplier Attribute

5823 *Multiplier* provides a value to be multiplied against a raw or uncompensated sensor count of
5824 Energy, Gas, or Water being measured by the metering device. If present, this attribute must be
5825 applied against all summation, consumption and demand values to derive the delivered and
5826 received values expressed in the unit of measure specified. This attribute must be used in
5827 conjunction with the *Divisor* attribute.

5828 D.3.2.2.4.3 Divisor Attribute

5829 *Divisor* provides a value to divide the results of applying the *Multiplier Attribute* against a raw
5830 or uncompensated sensor count of Energy, Gas, or Water being measured by the metering
5831 device. If present, this attribute must be applied against all summation, consumption and
5832 demand values to derive the delivered and received values expressed in the unit of measure
5833 specified. This attribute must be used in conjunction with the *Multiplier* attribute.

5834 D.3.2.2.4.4 SummationFormatting Attribute

5835 *SummationFormatting* provides a method to properly decipher the number of digits and the
5836 decimal location of the values found in the Summation Information Set of attributes. This
5837 attribute is to be decoded as follows:

5838 **Bits 0 to 2:** Number of Digits to the right of the Decimal Point.

5839 **Bits 3 to 6:** Number of Digits to the left of the Decimal Point.

5840 **Bit 7:** If set, suppress leading zeros.

5841 This attribute shall be used against the following attributes:

- 5842 • *CurrentSummationDelivered*
- 5843 • *CurrentSummationReceived*
- 5844 • TOU Information attributes
- 5845 • *DFTSummation*
- 5846 • Block Information attributes

5847 D.3.2.2.4.5 DemandFormatting Attribute

5848 *DemandFormatting* provides a method to properly decipher the number of digits and the
5849 decimal location of the values found in the Demand-related attributes. This attribute is to be
5850 decoded as follows:

5851 **Bits 0 to 2:** Number of Digits to the right of the Decimal Point.

5852 **Bits 3 to 6:** Number of Digits to the left of the Decimal Point.

5853 **Bit 7:** If set, suppress leading zeros.

5854 This attribute shall be used against the following attributes:

5855 • *CurrentMaxDemandDelivered*

5856 • *CurrentMaxDemandReceived*

5857 • *InstantaneousDemand*

5858 D.3.2.2.4.6 HistoricalConsumptionFormatting Attribute

5859 *HistoricalConsumptionFormatting* provides a method to properly decipher the number of digits
5860 and the decimal location of the values found in the Historical Consumption Set of attributes. This
5861 attribute is to be decoded as follows:

5862 **Bits 0 to 2:** Number of Digits to the right of the Decimal Point.

5863 **Bits 3 to 6:** Number of Digits to the left of the Decimal Point.

5864 **Bit 7:** If set, suppress leading zeros.

5865 This attribute shall be used against the following attributes:

5866 • *CurrentDayConsumptionDelivered*

5867 • *CurrentDayConsumptionReceived*

5868 • *PreviousDayConsumptionDelivered*

5869 • *PreviousDayConsumptionReceived*

5870 • *CurrentPartialProfileIntervalValue*

5871 • *Intervals*

5872 • *DailyConsumptionTarget*

5873 • *CurrentDayConsumptionDelivered*

5874 • *CurrentDayConsumptionReceived*

- 5875 • *PreviousDayNConsumptionDelivered*
 5876 • *PreviousDayNConsumptionReceived*
 5877 • *CurrentWeekConsumptionDelivered*
 5878 • *CurrentWeekConsumptionReceived*
 5879 • *PreviousWeekNConsumptionDelivered*
 5880 • *PreviousWeekNConsumptionReceived*
 5881 • *CurrentMonthConsumptionDelivered*
 5882 • *CurrentMonthConsumptionReceived*
 5883 • *PreviousMonthNConsumptionDelivered*
 5884 • *PreviousMonthNConsumptionReceived*

5885 D.3.2.2.4.7 MeteringDeviceType Attribute

5886 *MeteringDeviceType* provides a label for identifying the type of metering device present. The
 5887 attribute are values representing Energy, Gas, Water, Thermal, Heat, Cooling, and mirrored
 5888 metering devices. The defined values are represented in Table D-27. (Note that these values
 5889 represent an Enumeration, and not an 8-bit BitMap as indicated in the attribute description.
 5890 For backwards compatibility reasons, the data type has not been changed, though the data
 5891 itself should be treated like an enum.)

5892 Where a mirror is provided for a battery-powered metering device, the mirror shall assume the
 5893 relevant 'Mirrored Metering' device type (127-139) whilst the meter itself shall utilize the
 5894 'Metering' device type (0 to 15). It shall be the responsibility of the device providing the
 5895 mirror to modify the Device Type shown on the mirror to that of a 'Mirrored Metering' device.

5896

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

5897 *Note: Heat and cooling meters are used for measurement and billing of heat (and cooling)*
 5898 *delivered through liquid (water) based central heating systems. The consumers are typically*
 5899 *billed by the kWh, calculated from the flow and the temperatures in and out.*

5900 D.3.2.2.4.8 SiteID Attribute

5901 The *SiteID* is a ZCL Octet String field capable of storing a 32 character string (the first Octet
 5902 indicates length) encoded in UTF-8 format. The *SiteID* is a text string, known in the UK as the
 5903 MPAN number for electricity, MPRN for gas and 'Stand Point' in South Africa. These numbers
 5904 specify the meter point location in a standardized way. The field is defined to accommodate
 5905 the number of characters typically found in the UK and Europe (16 digits). Generally speaking

5906 the field is numeric but is defined for the possibility of an alpha-numeric format by
5907 specifying an octet string.

5908 D.3.2.2.4.9 MeterSerialNumber Attribute

5909 The *MeterSerialNumber* is a ZCL Octet String field capable of storing a 24 character string
5910 (the first Octet indicates length) encoded in UTF-8 format. It is used to provide a unique
5911 identification of the metering device.

5912 D.3.2.2.4.10 EnergyCarrierUnitOfMeasure Attribute

5913 The *EnergyCarrierUnitOfMeasure* specifies the unit of measure that the *EnergyCarrier* is
5914 measured in. This unit of measure is typically a unit of volume or flow and cannot be an
5915 amount of energy. The enumeration of this attribute is otherwise identical to the *UnitofMeasure*
5916 attribute (Table D-26).

5917 D.3.2.2.4.11 EnergyCarrierSummationFormatting Attribute

5918 *EnergyCarrierSummationFormatting* provides a method to properly decipher the number of
5919 digits and the decimal location of the values found in the Summation- related attributes.

5920 This attribute is to be decoded as follows:

5921 **Bits 0 to 2:** Number of Digits to the right of the Decimal Point.

5922 **Bits 3 to 6:** Number of Digits to the left of the Decimal Point.

5923 **Bit 7:** If set, suppress leading zeros.

5924 This attribute shall be used in relation with the following attributes:

- 5925 • *CurrentInletEnergyCarrierSummation*

- 5926 • *CurrentOutletEnergyCarrierSummation*

5927 D.3.2.2.4.12 EnergyCarrierDemandFormatting Attribute

5928 *EnergyCarrierDemandFormatting* provides a method to properly decipher the number of
5929 digits and the decimal location of the values found in the Demand- related attributes.

5930 This attribute is to be decoded as follows:

5931 **Bits 0 to 2:** Number of Digits to the right of the Decimal Point.

5932 **Bits 3 to 6:** Number of Digits to the left of the Decimal Point.

5933 **Bit 7:** If set, suppress leading zeros.

5934 This attribute shall be used in relation with the following attributes:

- 5935 • *CurrentInletEnergyCarrierDemand*
 5936 • *CurrentOutletEnergyCarrierDemand*
 5937 • *CurrentDayMaxEnergyCarrierDemand*
 5938 • *PreviousDayMaxEnergyCarrierDemand*
 5939 • *CurrentMonthMaxEnergyCarrierDemand*
 5940 • *CurrentMonthMinEnergyCarrierDemand*
 5941 • *CurrentYearMinEnergyCarrierDemand*
 5942 • *CurrentYearMaxEnergyCarrierDemand*

5943 D.3.2.2.4.13 TemperatureUnitOfMeasure Attribute

5944 The *TemperatureUnitOfMeasure* specifies the unit of measure that temperatures are measured
 5945 in. The enumeration of this attribute is as follows.

5946 **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

5947

5948 D.3.2.2.4.14 TemperatureFormatting Attribute

5949 *TemperatureFormatting* provides a method to properly decipher the number of digits and the
 5950 decimal location of the values found in the Temperature-related attributes. This attribute is to
 5951 be decoded as follows:

5952 **Bits 0 to 2:** Number of Digits to the right of the Decimal Point.

5953 **Bits 3 to 6:** Number of Digits to the left of the Decimal Point.

5954 **Bit 7:** If set, suppress leading zeros.

5955 This attribute shall be used in relation with the following attributes:

5956 • *InletTemperature*

5957 • *OutletTemperature*

5958 • *ControlTemperature*

5959 D.3.2.2.4.15 ModuleSerialNumber Attribute

5960 The *ModuleSerialNumber* attribute represents the serial number (unique identifier) of the meter
5961 module. It is a ZCL Octet String field capable of storing a 24 character string (the first Octet
5962 indicates length) encoded in UTF-8 format. It shall be used to uniquely identify the meter
5963 communications module.

5964 D.3.2.2.4.16 OperatingTariffLabelDelivered Attribute

5965 The *OperatingTariffLabelDelivered* attribute is the meter's version of the *TariffLabel* attribute
5966 that is found within the Tariff Information attribute set of the Price Cluster. It is used to identify
5967 the current consumption tariff operating on the meter. See section D.4.2.2.7.1. The attribute is a
5968 ZCL Octet String field capable of storing a 24 character string (the first Octet indicates
5969 length) encoded in UTF-8 format.

5970 D.3.2.2.4.17 OperatingTariffLabelReceived Attribute

5971 The *OperatingTariffLabelReceived* attribute is the meter's version of the *ReceivedTariffLabel*
5972 attribute that is found within the Tariff Information attribute set of the Price Cluster. It is used to
5973 identify the current generation tariff operating on the meter. See section D.4.2.2.15.1. The
5974 attribute is a ZCL Octet String field capable of storing a 24 character string (the first Octet
5975 indicates length) encoded in UTF-8 format.

5976 D.3.2.2.4.18 CustomerIDNumber Attribute

5977 The *CustomerIDNumber* attribute provides a customer identification which may be used to
5978 confirm the customer at the premises. The attribute is a ZCL Octet String field capable of storing
5979 a 24 character string (not including the first Octet which indicates length) encoded in UTF-8
5980 format.

5981 D.3.2.2.4.19 AlternativeUnitofMeasure Attribute

5982 Unless stated otherwise, the *AlternativeUnitofMeasure* attribute provides a base for the attributes
5983 in the Alternative Historical Consumption attribute set defined in Table D-46.

5984 The *AlternativeUnitofMeasure* attribute shall be supported if any of the attributes within the
5985 Alternative Historical Consumption attribute set are to be used.

5986 The *AlternativeUnitofMeasure* attribute shall be set to a value that is different to the
5987 *UnitOfMeasure* attribute.

5988 The *AlternativeUnitofMeasure* attribute is an 8-bit enumerated field. The possible values for this
5989 attribute are listed in Table D-26.

5990 D.3.2.2.4.20 AlternativeDemandFormatting Attribute

5991 *AlternativeDemandFormatting* provides a method to properly decipher the number of digits and
5992 the decimal location of the values found in the Alternative Demand-related attributes. This
5993 attribute is to be decoded as follows:

5994 **Bits 0 to 2:** Number of Digits to the right of the Decimal Point.

5995 **Bits 3 to 6:** Number of Digits to the left of the Decimal Point.

5996 **Bit 7:** If set, suppress leading zeros.

5997 This attribute shall be used against the following attribute:

5998 • *AlternativeInstantaneousDemand*

5999

6000 D.3.2.2.4.21 AlternativeConsumptionFormatting Attribute

6001 *AlternativeConsumptionFormatting* provides a method to properly decipher the number of digits
6002 and the decimal location of the consumption values found in the Alternative Historical
6003 Consumption Set of attributes. This attribute is to be decoded as follows:

6004 **Bits 0 to 2:** Number of Digits to the right of the Decimal Point.

6005 **Bits 3 to 6:** Number of Digits to the left of the Decimal Point.

6006 **Bit 7:** If set, suppress leading zeros.

6007 This attribute shall be used against the following attributes:

6008 • *CurrentDayAlternativeConsumptionDelivered*

6009 • *CurrentDayAlternativeConsumptionReceived*

6010 • *PreviousDayAlternativeConsumptionDelivered*

6011 • *PreviousDayAlternativeConsumptionReceived*

6012 • *CurrentAlternativePartialProfileIntervalValue*

6013 • *PreviousDayNALternativeConsumptionDelivered*

6014 • *PreviousDayNALternativeConsumptionReceived*

6015 • *CurrentWeekAlternativeConsumptionDelivered*

- 6016 • *CurrentWeekAlternativeConsumptionReceived*
 6017 • *PreviousWeekNAlternativeConsumptionDelivered*
 6018 • *PreviousWeekNAlternativeConsumptionReceived*
 6019 • *CurrentMonthAlternativeConsumptionDelivered*
 6020 • *CurrentMonthAlternativeConsumptionReceived*
 6021 • *PreviousMonthNAlternativeConsumptionDelivered*
 6022 • *PreviousMonthNAlternativeConsumptionReceived*

6023

6024

6025 D.3.2.2.5 **Historical Consumption Attribute**

6026 The Historical Consumption attribute set allows historical information to be presented in a base
 6027 defined by the *UnitofMeasure* attribute (see D.3.2.2.4.1). The attributes within this set are
 6028 defined in Table D-29.

6029 *Note: Historical Consumption Attributes 0x0409-0x040E, 0x0411 and 0x0412 in this revision of*
 6030 *this specification are provisional and not certifiable. This feature set may change before*
 6031 *reaching certifiable status in a future revision of this specification.*

6032

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 0x173C	Read Only	0x0000	O
0x045D to 0x04FF	Reserved					

6033

6034 D.3.2.2.5.1 InstantaneousDemand Attribute

6035 *InstantaneousDemand* represents the current Demand of Energy, Gas, or Water delivered or
 6036 received at the premises. Positive values indicate demand delivered to the premises where
 6037 negative values indicate demand received from the premises. *InstantaneousDemand* is
 6038 updated continuously as new measurements are made. The frequency of updates to this field is

6039 specific to the metering device, but should be within the range of once every second to once every
6040 5 seconds.

6041 D.3.2.2.5.2 CurrentDayConsumptionDelivered Attribute

6042 *CurrentDayConsumptionDelivered* represents the summed value of Energy, Gas, or Water
6043 delivered to the premises since the *Historical Freeze Time* (HFT). If optionally provided,
6044 *CurrentDayConsumptionDelivered* is updated continuously as new measurements are made. If
6045 the optional HFT attribute is not available, default to midnight local time.

6046 D.3.2.2.5.3 CurrentDayConsumptionReceived Attribute

6047 *CurrentDayConsumptionReceived* represents the summed value of Energy, Gas, or Water
6048 received from the premises since the *Historical Freeze Time* (HFT). If optionally provided,
6049 *CurrentDayConsumptionReceived* is updated continuously as new measurements are made. If
6050 the optional HFT attribute is not available, default to midnight local time.

6051 D.3.2.2.5.4 PreviousDayConsumptionDelivered Attribute

6052 *PreviousDayConsumptionDelivered* represents the summed value of Energy, Gas, or Water
6053 delivered to the premises within the previous 24 hour period starting at the *Historical Freeze*
6054 *Time* (HFT). If optionally provided, *PreviousDayConsumptionDelivered* is updated every HFT.
6055 If the optional HFT attribute is not available, default to midnight local time.

6056 D.3.2.2.5.5 PreviousDayConsumptionReceived Attribute

6057 *PreviousDayConsumptionReceived* represents the summed value of Energy, Gas, or Water
6058 received from the premises within the previous 24 hour period starting at the *Historical Freeze*
6059 *Time* (HFT). If optionally provided, *PreviousDayConsumptionReceived* is updated every HFT. If
6060 the optional HFT attribute is not available, default to midnight local time.

6061 D.3.2.2.5.6 CurrentPartialProfileIntervalStartTimeDelivered Attribute

6062 *CurrentPartialProfileIntervalStartTimeDelivered* represents the start time of the current Load
6063 Profile interval being accumulated for commodity delivered.

6064 D.3.2.2.5.7 CurrentPartialProfileIntervalStartTimeReceived Attribute

6065 *CurrentPartialProfileIntervalStartTimeReceived* represents the start time of the current Load
6066 Profile interval being accumulated for commodity received.

6067 D.3.2.2.5.8 CurrentPartialProfileIntervalValueDelivered Attribute

6068 *CurrentPartialProfileIntervalValueDelivered* represents the value of the current Load Profile
6069 interval being accumulated for commodity delivered.

6070 D.3.2.2.5.9 CurrentPartialProfileIntervalValueReceived Attribute

6071 *CurrentPartialProfileIntervalValueReceived* represents the value of the current Load Profile
6072 interval being accumulated for commodity received.

6073 D.3.2.2.5.10 CurrentDayMaxPressure Attribute

6074 *CurrentDayMaxPressure* is the maximum pressure reported during a day from the water or gas
6075 meter.

6076 D.3.2.2.5.11 PreviousDayMaxPressure Attribute

6077 *PreviousDayMaxPressure* represents the maximum pressure reported during previous day from
6078 the water or gas meter.

6079 D.3.2.2.5.12 CurrentDayMinPressure Attribute

6080 *CurrentDayMinPressure* is the minimum pressure reported during a day from the water or gas
6081 meter.

6082 D.3.2.2.5.13 PreviousDayMinPressure Attribute

6083 *PreviousDayMinPressure* represents the minimum pressure reported during previous day from
6084 the water or gas meter.

6085 D.3.2.2.5.14 CurrentDayMaxDemand Attribute

6086 *CurrentDayMaxDemand* represents the maximum demand or rate of delivered value of
6087 Energy, Gas, or Water being utilized at the premises.

6088 D.3.2.2.5.15 PreviousDayMaxDemand Attribute

6089 *PreviousDayMaxDemand* represents the maximum demand or rate of delivered value of
6090 Energy, Gas, or Water being utilized at the premises.

6091 *Note: At the end of a day the metering device will transfer the CurrentDayMaxPressure into*
6092 *PreviousDayMaxPressure, CurrentDayMinPressure into PreviousDayMinPressure and*
6093 *CurrentDayMaxDemand into PreviousDayMaxDemand.*

6094 D.3.2.2.5.16 CurrentMonthMaxDemand Attribute

6095 *CurrentMonthMaxDemand* is the maximum demand reported during a month from the meter.

6096 For electricity, heat and cooling meters this is the maximum power reported in a month.

6097 D.3.2.2.5.17 CurrentYearMaxDemand Attribute

6098 *CurrentYearMaxDemand* is the maximum demand reported during a year from the meter.

6099 For electricity, heat and cooling meters this is the maximum power reported in a year.

6100 D.3.2.2.5.18 CurrentDayMaxEnergyCarrierDemand Attribute

6101 *CurrentDayMaxEnergyCarrierDemand* is the maximum energy carrier demand reported during
6102 a day from the meter.

6103 **Note:** At the end of a day the meter will transfer the *CurrentDayMaxEnergyCarrierDemand* into
6104 *PreviousDayMaxEnergyCarrierDemand*.

6105 For heat and cooling meters this is the maximum flow rate on the inlet reported in a day.

6106 D.3.2.2.5.19 PreviousDayMaxEnergyCarrierDemand Attribute

6107 *PreviousDayMaxEnergyCarrierDemand* is the maximum energy carrier demand reported during
6108 the previous day from the meter.

6109 D.3.2.2.5.20 CurrentMonthMaxEnergyCarrierDemand Attribute

6110 *CurrentMonthMaxEnergyCarrierDemand* is the maximum energy carrier demand reported during
6111 a month from the meter.

6112 For heat and cooling meters this is the maximum flow rate on the inlet reported in a month.

6113 D.3.2.2.5.21 CurrentMonthMinEnergyCarrierDemand Attribute

6114 *CurrentMonthMinEnergyCarrierDemand* is the minimum energy carrier demand reported during
6115 a month from the meter.

6116 For heat and cooling meters this is the minimum flow rate on the inlet reported in a month.

6117 **Note:** This attribute may be used to detect leaks if there has been no flow rate of zero in the last
6118 month.

6119 D.3.2.2.5.22 CurrentYearMaxEnergyCarrierDemand Attribute

6120 *CurrentYearMaxEnergyCarrierDemand* is the maximum energy carrier demand reported during
6121 a year from the meter.

6122 For heat and cooling meters this is the maximum flow rate on the inlet reported in a year.

6123 D.3.2.2.5.23 CurrentYearMinEnergyCarrierDemand Attribute

6124 *CurrentYearMinEnergyCarrierDemand* is the minimum energy carrier demand reported during
6125 a year from the heat meter.

6126 For heat and cooling meters this is the minimum flow rate on the inlet reported in a year.

6127 **Note:** This attribute may be used to detect leaks if there has been no flow rate of zero in the last
6128 year

6129 D.3.2.2.5.24 PreviousDayNConsumptionDelivered Attribute

6130 *PreviousDayNConsumptionDelivered* represents the summed value of Energy, Gas, or Water
6131 delivered to 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.3.2.2.5.25 PreviousDayNConsumptionReceived Attribute

6134 *PreviousDayNConsumptionReceived* represents the summed value of Energy, Gas, or Water
6135 received from the premises within the previous 24 hour period starting at the *Historical Freeze*
6136 *Time* (HFT). If the optional HFT attribute is not available, default to midnight local time.

6137 D.3.2.2.5.26 CurrentWeekConsumptionDelivered Attribute

6138 *CurrentWeekConsumptionDelivered* represents the summed value of Energy, Gas, or Water
6139 delivered to the premises since the *Historical Freeze Time* (HFT) on Monday to the last HFT
6140 read. If optionally provided, *CurrentWeekConsumptionDelivered* is updated continuously as new
6141 measurements are made. If the optional HFT attribute is not available, default to midnight local
6142 time.

6143 D.3.2.2.5.27 CurrentWeekConsumptionReceived Attribute

6144 *CurrentWeekConsumptionReceived* represents the summed value of Energy, Gas, or Water
6145 received from the premises since the *Historical Freeze Time* (HFT) on Monday to the last HFT
6146 read. If optionally provided, *CurrentWeekConsumptionReceived* is updated continuously as new
6147 measurements are made. If the optional HFT attribute is not available, default to midnight local
6148 time.

6149 D.3.2.2.5.28 PreviousWeekNConsumptionDelivered Attribute

6150 *PreviousWeekNConsumptionDelivered* represents the summed value of Energy, Gas, or Water
6151 delivered to the premises within the previous week period starting at the *Historical Freeze Time*
6152 (*HFT*) on the Monday to the Sunday. If the optional HFT attribute is not available, default to
6153 midnight local time.

6154 D.3.2.2.5.29 PreviousWeekNConsumptionReceived Attribute

6155 *PreviousWeekNConsumptionReceived* represents the summed value of Energy, Gas, or Water
6156 received from the premises within the previous week period starting at the *Historical Freeze*
6157 *Time* (*HFT*) on the Monday to the Sunday. If the optional HFT attribute is not available, default
6158 to midnight local time.

6159 D.3.2.2.5.30 CurrentMonthConsumptionDelivered Attribute

6160 *CurrentMonthConsumptionDelivered* represents the summed value of Energy, Gas, or Water
6161 delivered to the premises since the *Historical Freeze Time* (*HFT*) on the 1st of the month to the
6162 last HFT read. If optionally provided, *CurrentMonthConsumptionDelivered* is updated

6163 continuously as new measurements are made. If the optional HFT attribute is not available,
 6164 default to midnight local time.

6165 D.3.2.2.5.31 CurrentMonthConsumptionReceived Attribute

6166 *CurrentMonthConsumptionReceived* represents the summed value of Energy, Gas, or Water
 6167 received from the premises since the *Historical Freeze Time* (HFT) on the 1st of the month to the
 6168 last HFT read. If optionally provided, *CurrentMonthConsumptionReceived* is updated
 6169 continuously as new measurements are made. If the optional HFT attribute is not available,
 6170 default to midnight local time.

6171 D.3.2.2.5.32 PreviousMonthNConsumptionDelivered Attribute

6172 *PreviousMonthNConsumptionDelivered* represents the summed value of Energy, Gas, or Water
 6173 delivered to the premises within the previous Month period starting at the *Historical Freeze Time*
 6174 (HFT) on the 1st of the month to the last day of the month. If the optional HFT attribute is not
 6175 available, default to midnight local time.

6176 D.3.2.2.5.33 PreviousMonthNConsumptionReceived Attribute

6177 *PreviousMonthNConsumptionReceived* represents the summed value of Energy, Gas, or Water
 6178 received from the premises within the previous month period starting at the *Historical Freeze Time*
 6179 (HFT) on the 1st of the month to the last day of the month. If the optional HFT attribute is not
 6180 available, default to midnight local time.

6181 D.3.2.2.5.34 HistoricalFreezeTime Attribute

6182 *HistoricalFreezeTime* (HFT) represents the time of day, in Local Time, when Historical
 6183 Consumption attributes and/or Alternative Historical Consumption attributes are captured.
 6184 *HistoricalFreezeTime* is an unsigned 16-bit value representing the hour and minutes for HFT.
 6185 The byte usages are:

6186 **Bits 0 to 7:** Range of 0 to 0x3B representing the number of minutes past the top of the hour.

6187

6188 **Bits 8 to 15:** Range of 0 to 0x17 representing the hour of the day (in 24-hour format).

6189

6190 D.3.2.2.6 Load Profile Configuration

6191 The Load Profile Configuration Attribute Set is defined in Table D-30.

6192 **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					
---------------------	----------	--	--	--	--	--

6193

6194 D.3.2.2.6.1 MaxNumberOfPeriodsDelivered Attribute

6195 *MaxNumberofPeriodsDelivered* represents the maximum number of intervals the device is
 6196 capable of returning in one *Get Profile Response* command. It is required
 6197 *MaxNumberofPeriodsDelivered* fit within the default Fragmentation ASDU size of 128 bytes,
 6198 or an optionally agreed upon larger Fragmentation ASDU size supported by both devices.
 6199 Please refer to sub-clause 5.3.8 for further details on Fragmentation settings.

6200

6201 D.3.2.2.7 Supply Limit Attributes

6202 This set of attributes is used to implement a “Supply Capacity Limit” program where the
 6203 demand at the premises is limited to a preset consumption level over a preset period of time.
 6204 Should this preset limit be exceeded the meter could interrupt supply to the premises or to
 6205 devices within the premises. The supply limit information in this attribute set can be used by
 6206 In-Home displays, PCTs, or other devices to display a warning when the supply limit is
 6207 being approached. The Supply Limit Attribute Set is defined in Table D-31.

6208

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					

6209

6210 D.3.2.2.7.1 CurrentDemandDelivered Attribute

6211 *CurrentDemandDelivered* represents the current Demand of Energy, Gas, or Water delivered at
 6212 the premises. *CurrentDemandDelivered* may be continuously updated as new measurements are
 6213 acquired, but at a minimum *CurrentDemandDelivered* must be updated at the end of each
 6214 integration sub-period, which can be obtained by dividing the *DemandIntegrationPeriod* by
 6215 the *NumberOfDemandSubintervals*.

6216 This attribute shall be adjusted using the *Multiplier* and *Divisor* attributes found in the Formatting
 6217 Attribute Set and can be formatted using the *DemandFormatting* attribute. The final result
 6218 represents an engineering value in the unit defined by the *UnitofMeasure* attribute.

6219 D.3.2.2.7.2 DemandLimit Attribute

6220 *DemandLimit* reflects the current supply demand limit set in the meter. This value can be
 6221 compared to the *CurrentDemandDelivered* attribute to understand if limits are being approached
 6222 or exceeded.

6223 Adjustment and formatting of this attribute follow the same rules as the
 6224 *CurrentDemandDelivered*.

6225 A value of “0xFFFF” indicates “demand limiting” is switched off.

6226 D.3.2.2.7.3 DemandIntegrationPeriod Attribute

6227 *DemandIntegrationPeriod* is the number of minutes over which the *CurrentDemandDelivered*
 6228 attribute is calculated. Valid range is 0x01 to 0xFF. 0x00 is a reserved value.

6229 D.3.2.2.7.4 NumberOfDemandSubintervals Attribute

6230 *NumberOfDemandSubintervals* represents the number of subintervals used within the
 6231 *DemandIntegrationPeriod*. The subinterval duration (in minutes) is obtained by dividing the
 6232 *DemandIntegrationPeriod* by the *NumberOfDemandSubintervals*. The *CurrentDemandDelivered*
 6233 attribute is updated at the end of each subinterval. Valid range is 0x01 to 0xFF. 0x00 is a
 6234 reserved value.

6235 As a Rolling Demand example, *DemandIntegrationPeriod* could be set at 30 (for 30 minute
6236 period) and *NumberOfDemandSubintervals* could be set for 6. This would provide 5 minute
6237 (30/6 = 5) subinterval periods.

6238 As a Block Demand example, *DemandIntegrationPeriod* could be set at 30 (for 30 minute
6239 period) and *NumberOfDemandSubintervals* could be set for 1. This would provide a single 30
6240 minute subinterval period.

6241 D.3.2.2.7.5 DemandLimitArmDuration Attribute

6242 An unsigned 16-bit integer that defines the length of time, in seconds, that the supply shall be
6243 disconnected if the *DemandLimit* attribute is enabled and the limit is exceeded. At the end of the
6244 time period the meter shall move to the *ARMED* status. This will allow the user to reconnect the
6245 supply.

6246 D.3.2.2.7.6 LoadLimitSupplyState Attribute

6247 The *LoadLimitSupplyState* attribute indicates the required status of the supply once device is in a
6248 load limit state. The enumerated values for this field are outlined in Table D-68..

6249 D.3.2.2.7.7 LoadLimitCounter Attribute

6250 An unsigned 8-bit integer used for counting the number of times that the demand limit has
6251 exceeded the set threshold.

6252 This attribute shall be reset to zero on receipt of a *ResetLoadLimitCounter* command (see
6253 D.3.3.3.1.11 for further details).

6254 D.3.2.2.7.8 SupplyTamperState Attribute

6255 The *SupplyTamperState* indicates the required status of the supply following the detection of a
6256 tamper event within the metering device. The enumerated values for this field are outlined in
6257 Table D-68.

6258 D.3.2.2.7.9 SupplyDepletionState Attribute

6259 The *SupplyDepletionState* indicates the required status of the supply following detection of a
6260 depleted battery within the metering device. The enumerated values for this field are outlined in
6261 Table D-68.

6262 D.3.2.2.7.10 SupplyUncontrolledFlowState Attribute

6263 The *SupplyUncontrolledFlowState* indicates the required status of the supply following detection
6264 of an uncontrolled flow event within the metering device. The enumerated values for this field
6265 are outlined in Table D-68.

6266

6267 D.3.2.2.8 **Block Information Attribute Set (Delivered)**

6268 The following set of attributes provides a remote access to the Electric, Gas, or Water metering
 6269 device's block readings. The Block Information attribute set supports Block pricing and
 6270 combined Tier-Block pricing, the number of blocks is one greater than the number of block
 6271 thresholds defined in the Pricing cluster.

6272 This attribute set is ONLY for Energy, Gas or Water delivered to and consumed within the
 6273 premises.

6274 ***Note:*** *Block Information Attribute Set in this revision of this specification are provisional and
 6275 not certifiable. This feature set may change before reaching certifiable status in a future
 6276 revision of this specification.*

6277 **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>CurrentTier15 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

6278

6279 D.3.2.2.8.1 CurrentTierNBlockNSummationDelivered Attributes

6280 Attributes *CurrentNoTierBlock1SummationDelivered* through
 6281 *CurrentTier15Block16SummationDelivered* represent the most recent summed value of Energy,
 6282 Gas, or Water delivered to the premises (i.e delivered to the customer from the utility) at a
 6283 specific price tier as defined by a TOU schedule, Block Threshold or a real time pricing period.
 6284 If optionally provided, attributes *CurrentNoTierBlock1SummationDelivered* through
 6285 *CurrentTier15Block16SummationDelivered* are updated continuously as new measurements are
 6286 made.

6287 **Note:** *SummationFormatting shall be used against the Block Information attribute set. The*
 6288 *expected practical limit for the number of Block attributes supported is 64. The*
 6289 *CurrentTierNBlockNSummationDelivered attributes are reset at the start of each Block*
 6290 *Threshold Period.*

6291 D.3.2.2.9 Alarms Set

6292 The following set of attributes provides a means to control which alarms may be generated from
 6293 the meter.

6294 **Note:** *Alarms Attribute Set in this revision of this specification are provisional and not*
 6295 *certifiable. This feature set may change before reaching certifiable status in a future revision of*
 6296 *this specification.*

6297 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					

6298

6299 D.3.2.2.9.1 AlarmMask Attributes

6300 The *AlarmMask* attributes of the Alarm Attribute Set specify whether each of the alarms listed in
 6301 the corresponding alarm group in Table D-34 through Table D-42 is enabled. When the bit
 6302 number corresponding to the alarm number (minus the group offset) is set to 1, the alarm is
 6303 enabled, else it is disabled. Bits not corresponding to a code in the respective table are reserved.

6304 D.3.2.2.9.2 Alarm Codes

6305 The alarm codes are organized in logical groups corresponding to the meter type as listed below.
 6306 The three main alarm groups are: Generic, Electricity, and Flow/ Pressure. The Flow/Pressure
 6307 Alarm Group is further divided into Generic Flow/ Pressure, Water Specific, Heat and Cooling
 6308 Specific, and Gas Specific. It is left for the manufacturer to select which (if any) alarm codes to
 6309 support.

6310

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

6311
 6312 The generic Alarm Group maps the status from the *MeterStatus* attribute into a corresponding
 6313 alarm. Hence, depending on the meter type, an alarm belonging to the Generic Alarm Group
 6314 may have a different meaning. See sub-clause D.3.2.2.3. In the case of overlap of alarm codes
 6315 from the Generic Alarm Group with codes in other groups, e.g. Burst Detect, it is
 6316 recommended to only use the code of the Generic Alarm Group.

6317

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

6318
 6319 The Electricity Alarm Group defines alarms specific for electricity meters as defined below.

6320

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

6321

6322 The Generic Flow/Pressure Alarm Group defines alarms specific for Flow/Pressure based meters
 6323 i.e. Water, Heat, Cooling, or Gas meters as defined below.

6324

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

6325

6326 The Water Specific Alarm Group defines alarms specific for Water meters as defined below.

6327

Table D-38– Water Specific Alarm Group

Alarm Code	Alarm Condition
40-4F	Reserved

6328

6329 The Heat and Cooling Specific Alarm Group defines alarms specific for Heat or Cooling meters
6330 as defined below.

6331

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

6332

6333 The Gas Specific Alarm Group defines alarms specific for Gas meters as defined below.

6334

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

6335

6336 The Extended Generic Alarm Group is an additional set of generic meter alarms.

6337

Table D-41– Extended Generic Alarm Group

Alarm Code	Alarm Condition
0x70	Measurement System Error

Alarm Code	Alarm Condition
0x71	Watchdog Error
0x72	Supply Disconnect Failure
0x73	Supply Connect Failure
0x74	Measurment Software Changed
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

6339 The Manufacturer Specific Alarm Group defines alarms specific for any meters as defined
 6340 below, these are used for meter specific functionality that is not covered by the current smart
 6341 energy specification.

6342

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

6343

6344 **D.3.2.2.10 Block Information Attribute Set (Received)**

6345 The following set of attributes provides a remote access to the Electric, Gas, or Water metering
 6346 devices block readings. The Block Information attribute set supports Block pricing and combined
 6347 Tier-Block pricing, the number of blocks is one greater than the number of block thresholds
 6348 defined in the Pricing cluster.

6349 This attribute set is ONLY for Energy generated from the premises and received by the utility.

6350

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

6351

6352 D.3.2.2.10.1 CurrentTierNBlockNSummationReceived Attributes

6353 Attributes *CurrentNoTierBlock1SummationReceived* through
 6354 *CurrentTier15Block16SummationReceived* represent the most recent summed value of Energy,
 6355 Gas, or Water received from the premises (i.e. delivered to the utility from the customer) at a
 6356 specific price tier as defined by a TOU schedule, Block Threshold or a real time pricing period. If
 6357 optionally provided, attributes *CurrentNoTierBlock1SummationReceived* through

6358 *CurrentTier15Block16SummationReceived* are updated continuously as new measurements are
 6359 made.

6360 **Note:** *SummationFormatting* shall be used against the Block Information attribute set. The
 6361 practical limit for the number of Block attributes supported is 32. The
 6362 *CurrentTierNBlockNSummationReceived* attributes are reset at the start of each Block Threshold
 6363 Period.

6364 **D.3.2.2.11 Meter Billing Attribute Set**

6365 The billing information within this attribute set is created on the metering device. The
 6366 information in this attribute set is intended for use by simple IHDs.

6367 **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					

6369

6370 D.3.2.2.11.1 BillToDateDelivered Attribute

6371 *BillToDateDelivered* provides a value for the costs in the current billing period. This attribute is
6372 measured in a base unit of Currency with the decimal point located as indicated by the
6373 *BillDeliveredTrailingDigit* attribute.

6374 D.3.2.2.11.2 BillToDateTimeStampDelivered Attribute

6375 The UTC timestamp when the associated *BillToDateDelivered* attribute was last updated.

6376 D.3.2.2.11.3 ProjectedBillDelivered Attribute

6377 *ProjectedBillDelivered* provides a value indicating what the estimated state of the account will be
6378 at the end of the billing period based on past consumption. This attribute is measured in a base
6379 unit of Currency with the decimal point located as indicated by the *BillDeliveredTrailingDigit*
6380 attribute.

6381 D.3.2.2.11.4 ProjectedBillTimeStampDelivered Attribute

6382 The UTC timestamp when the associated *ProjectedBillDelivered* attribute was last updated.

6383 D.3.2.2.11.5 BillDeliveredTrailingDigit Attribute

6384 An 8-bit BitMap used to determine where the decimal point is located in the
6385 *BillToDateDelivered* and *ProjectedBillDelivered* attributes. The most significant nibble
6386 indicates the number of digits to the right of the decimal point. The least significant nibble is
6387 reserved and shall be 0. The *BillDeliveredTrailingDigit* attribute represents the current active
6388 value.

6389 D.3.2.2.11.6 BillToDateReceived Attribute

6390 *BillToDateReceived* provides a value for the costs in the current billing period. This attribute is
6391 measured in a base unit of Currency with the decimal point located as indicated by the
6392 *BillReceivedTrailingDigit* attribute.

6393 D.3.2.2.11.7 BillToDateTimeStampReceived Attribute

6394 The UTC timestamp when the associated *BillToDateReceived* attribute was last updated.

6395 D.3.2.2.11.8 ProjectedBillReceived Attribute

6396 *ProjectedBillReceived* provides a value indicating what the estimated state of the account will be
6397 at the end of the billing period based on past generation. This attribute is measured in a base unit
6398 of Currency with the decimal point located as indicated by the *BillReceivedTrailingDigit*
6399 attribute.

6400 D.3.2.2.11.9 ProjectedBillTimeStampReceived Attribute

6401 The UTC timestamp when the associated *ProjectedBillReceived* attribute was last updated.

6402 D.3.2.2.11.10 BillReceivedTrailingDigit Attribute

6403 An 8-bit BitMap used to determine where the decimal point is located in the *BillToDateReceived*
 6404 and *ProjectedBillReceived* attributes. The most significant nibble indicates the number of digits
 6405 to the right of the decimal point. The least significant nibble is reserved and shall be 0. The
 6406 *BillReceivedTrailingDigit* attribute represents the current active value.

6407

6408 D.3.2.2.12 Supply Control Attribute Set

6409 **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					

6410

6411 D.3.2.2.12.1 ProposedChangeSupplyImplementationTime Attribute

6412 The *ProposedChangeImplementationTime* attribute indicates the time at which a proposed
 6413 change to the supply is to be implemented. If there is no change of supply pending, this attribute
 6414 will be set to 0xFFFFFFFF.

6415 D.3.2.2.12.2 ProposedChangeSupplyStatus Attribute

6416 The *ProposedChangeSupplyStatus* indicates the proposed status of the supply once the change to
6417 the supply has been implemented. The enumerated values of this field are outlined in Table
6418 D-56

6419 D.3.2.2.12.3 Uncontrolled Flow Threshold Attribute

6420 The *Uncontrolled Flow Threshold* attribute indicates the threshold above which a flow meter
6421 (e.g. Gas or Water) shall detect an uncontrolled flow. A value of 0x0000 indicates the feature in
6422 unused.

6423 D.3.2.2.12.4 Uncontrolled Flow Threshold Unit of Measure Attribute

6424 The *Uncontrolled Flow Threshold Unit of Measure* attribute indicates the unit of measure used in
6425 conjunction with the *Uncontrolled Flow Threshold* attribute. The enumeration used for this field
6426 shall match one of the *UnitOfMeasure* values using a pure binary format as defined in this cluster
6427 (see sub-clause D.3.2.2.4.1).

6428 D.3.2.2.12.5 Uncontrolled Flow Multiplier Attribute

6429 The *Uncontrolled Flow Multiplier* attribute indicates the multiplier, to be used in conjunction
6430 with the *Uncontrolled Flow Threshold* and *Uncontrolled Flow Divisor* attributes, to determine
6431 the true flow threshold value. A value of 0x0000 is not allowed.

6432 D.3.2.2.12.6 Uncontrolled Flow Divisor Attribute

6433 The *Uncontrolled Flow Divisor* attribute indicates the divisor, to be used in conjunction with the
6434 *Uncontrolled Flow Threshold* and *Uncontrolled Flow Multiplier* attributes, to determine the true
6435 flow threshold value. A value of 0x0000 is not allowed.

6436 D.3.2.2.12.7 Flow Stabilisation Period Attribute

6437 The *Flow Stabilisation Period* attribute indicates the time given to allow the flow to stabilize. It
6438 is defined in units of tenths of a second.

6439 D.3.2.2.12.8 Flow Measurement Period Attribute

6440 The *Flow Measurement Period* attribute indicates the period over which the flow is measured
6441 and compared against the *Uncontrolled Flow Threshold* attribute. It is defined in units of 1
6442 second.

6443

6444 D.3.2.2.13 Alternative Historical Consumption Attribute Set

6445 The Alternative Historical Attribute Set allows historical information to be presented in a base
6446 defined by the *AlternativeUnitofMeasure* (see D.3.2.2.4.19) and in a format defined by the
6447 *AlternativeDemandFormatting* and *AlternativeConsumptionFormatting* attributes (see

6448 D.3.2.2.4.20 and D.3.2.2.4.21 respectively). The attributes within this set are defined in Table
 6449 D-46.

6450

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>CurrentDayAlternativeConsumptionDelivered</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0C02	<i>CurrentDayAlternativeConsumptionReceived</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0C03	<i>PreviousDayAlternativeConsumptionDelivered</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0C04	<i>PreviousDayAlternativeConsumptionReceived</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0C05	<i>CurrentAlternativePartialProfileIntervalStartTimeDelivered</i>	UTCTime		Read Only	-	O
0x0C06	<i>CurrentAlternativePartialProfileIntervalStartTimeReceived</i>	UTCTime	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0C07	<i>CurrentAlternativePartialProfileIntervalValueDelivered</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0C08	<i>CurrentAlternativePartialProfileIntervalValueReceived</i>	Unsigned 24-bit Integer	0x000000 to 0xFFFFFFF	Read Only	-	O
0x0C09	<i>CurrentDayAlternativeMaxPressure</i>	Unsigned 48-bit Integer	0x000000 000000 to 0xFFFFFFF FFFFFFF	Read Only	-	O
0x0C0A	<i>CurrentDayAlternativeMinPressure</i>	Unsigned 48-bit Integer	0x000000 000000 to 0xFFFFFFF FFFFFFF	Read Only	-	O
0x0C0B	<i>PreviousDayAlternativeMaxPressure</i>	Unsigned 48-bit Integer	0x000000 000000 to 0xFFFFFFF FFFFFFF	Read Only	-	O

0x0C0C	<i>PreviousDay AlternativeMinPressure</i>	Unsigned 48-bit Integer	0x000000 000000 to 0xFFFFFFF FFFFFF	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	0x000000 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	0x000000000 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					

6452 D.3.2.2.13.1 AlternativeInstantaneousDemand Attribute

6453 *AlternativeInstantaneousDemand* represents the current Demand delivered or received at the
6454 premises. Positive values indicate demand delivered to the premises where negative values
6455 indicate demand received from the premises. *AlternativeInstantaneousDemand* is updated
6456 continuously as new measurements are made. The frequency of updates to this field is specific to
6457 the metering device, but should be within the range of once every second to once every 5 seconds.

6458 D.3.2.2.13.2 CurrentDayAlternativeConsumptionDelivered Attribute

6459 *CurrentDayAlternativeConsumptionDelivered* represents the summed value delivered to the
6460 premises since the *Historical Freeze Time* (HFT). If optionally provided,
6461 *CurrentDayAlternativeConsumptionDelivered* is updated continuously as new measurements are
6462 made. If the optional HFT attribute is not available, default to midnight local time.

6463 D.3.2.2.13.3 CurrentDayAlternativeConsumptionReceived Attribute

6464 *CurrentDayAlternativeConsumptionReceived* represents the summed value received from the
6465 premises since the *Historical Freeze Time* (HFT). If optionally provided,
6466 *CurrentDayAlternativeConsumptionReceived* is updated continuously as new measurements are
6467 made. If the optional HFT attribute is not available, default to midnight local time.

6468 D.3.2.2.13.4 PreviousDayAlternativeConsumptionDelivered Attribute

6469 *PreviousDayAlternativeConsumptionDelivered* represents the summed value delivered to the
6470 premises within the previous 24 hour period starting at the *lternative Historical Freeze Time*
6471 (HFT). If optionally provided, *PreviousDayAlternativeConsumptionDelivered* is updated every
6472 HFT. If the optional HFT attribute is not available, default to midnight local time.

6473 D.3.2.2.13.5 PreviousDayAlternativeConsumptionReceived Attribute

6474 *PreviousDayAlternativeConsumptionReceived* represents the summed value received from the
6475 premises within the previous 24 hour period starting at the *Historical Freeze Time* (HFT). If
6476 optionally provided, *PreviousDayAlternativeConsumptionReceived* is updated every HFT. If the
6477 optional HFT attribute is not available, default to midnight local time.

6478 D.3.2.2.13.6 CurrentAlternativePartialProfileIntervalStartTimeDelivered Attribute

6479 *CurrentAlternativePartialProfileIntervalStartTimeDelivered* represents the start time of the
6480 current Load Profile interval being accumulated for commodity delivered.

6481 D.3.2.2.13.7 CurrentAlternativePartialProfileIntervalStartTimeReceived Attribute

6482 *CurrentAlternativePartialProfileIntervalStartTimeReceived* represents the start time of the
6483 current Load Profile interval being accumulated for commodity received.

6484 D.3.2.2.13.8 CurrentAlternativePartialProfileIntervalValueDelivered Attribute

6485 *CurrentAlternativePartialProfileIntervalValueDelivered* represents the value of the current
6486 Load Profile interval being accumulated for commodity delivered.

6487 D.3.2.2.13.9 CurrentAlternativePartialProfileIntervalValueReceived Attribute

6488 *CurrentAlternativePartialProfileIntervalValueReceived* represents the value of the current
6489 Load Profile interval being accumulated for commodity received.

6490 D.3.2.2.13.10CurrentDayAlternativeMaxPressure Attribute

6491 *CurrentDayAlternativeMaxPressure* is the maximum pressure reported during a day from the
6492 water or gas meter.

6493 D.3.2.2.13.11PreviousDayAlternativeMaxPressure Attribute

6494 *PreviousDayAlternativeMaxPressure* represents the maximum pressure reported during previous
6495 day from the water or gas meter.

6496 D.3.2.2.13.12CurrentDayAlternativeMinPressure Attribute

6497 *CurrentDayAlternativeMinPressure* is the minimum pressure reported during a day from the
6498 water or gas meter.

6499 D.3.2.2.13.13PreviousDayAlternativeMinPressure Attribute

6500 *PreviousDayAlternativeMinPressure* represents the minimum pressure reported during previous
6501 day from the water or gas meter.

6502 D.3.2.2.13.14CurrentDayAlternativeMaxDemand Attribute

6503 *CurrentDayAlternativeMaxDemand* represents the maximum demand or rate of delivered
6504 value of Energy, Gas, or Water being utilized at the premises.

6505 D.3.2.2.13.15PreviousDayAlternativeMaxDemand Attribute

6506 *PreviousDayAlternativeMaxDemand* represents the maximum demand or rate of delivered
6507 value of Energy, Gas, or Water being utilized at the premises.

6508 *Note: At the end of a day the metering device will transfer the*
6509 *CurrentDayAlternativeMaxPressure into PreviousDayAlternativeMaxPressure,*
6510 *CurrentDayAlternativeMinPressure into PreviousDayAlternativeMinPressure and*
6511 *CurrentDayAlternativeMaxDemand into PreviousDayAlternativeMaxDemand.*

6512 D.3.2.2.13.16CurrentMonthAlternativeMaxDemand Attribute

6513 *CurrentMonthAlternativeMaxDemand* is the maximum demand reported during a month from
6514 the meter.

6515 For electricity, heat and cooling meters this is the maximum power reported in a month.

6516 D.3.2.2.13.17 CurrentYearAlternativeMaxDemand Attribute

6517 *CurrentYearAlternativeMaxDemand* is the maximum demand reported during a year from the meter.

6519 For electricity, heat and cooling meters this is the maximum power reported in a year.

6520 D.3.2.2.13.18 PreviousDayNAlternativeConsumptionDelivered Attribute

6521 *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.

6524 D.3.2.2.13.19 PreviousDayNAlternativeConsumptionReceived Attribute

6525 *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.

6528 D.3.2.2.13.20 CurrentWeekAlternativeConsumptionDelivered Attribute

6529 *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.

6534 D.3.2.2.13.21 CurrentWeekAlternativeConsumptionReceived Attribute

6535 *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.

6540 D.3.2.2.13.22 PreviousWeekNAlternativeConsumptionDelivered Attribute

6541 *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.

6545 D.3.2.2.13.23 PreviousWeekNAlternativeConsumptionReceived Attribute

6546 *PreviousWeekNAlternativeConsumptionReceived* represents the summed value received from the premises within the previous week period starting at the *Historical Freeze Time* (HFT) on the

6548 Monday to the Sunday. If the optional HFT attribute is not available, default to midnight local
 6549 time.

6550 D.3.2.2.13.24 CurrentMonthAlternativeConsumptionDelivered Attribute

6551 *CurrentMonthAlternativeConsumptionDelivered* represents the summed value delivered to the
 6552 premises since the *Historical Freeze Time* (HFT) on the 1st of the month to the last HFT read. If
 6553 optionally provided, *CurrentMonthAlternativeConsumptionDelivered* is updated continuously as
 6554 new measurements are made. If the optional HFT attribute is not available, default to midnight
 6555 local time.

6556 D.3.2.2.13.25 CurrentMonthAlternativeConsumptionReceived Attribute

6557 *CurrentMonthAlternativeConsumptionReceived* represents the summed value received from the
 6558 premises since the *Historical Freeze Time* (HFT) on the 1st of the month to the last HFT read. If
 6559 optionally provided, *CurrentMonthAlternativeConsumptionReceived* is updated continuously as
 6560 new measurements are made. If the optional HFT attribute is not available, default to midnight
 6561 local time.

6562 D.3.2.2.13.26 PreviousMonthNAlternativeConsumptionDelivered Attribute

6563 *PreviousMonthNAlternativeConsumptionDelivered* represents the summed value delivered to the
 6564 premises within the previous Month period starting at the *Historical Freeze Time* (HFT) on the
 6565 1st of the month to the last day of the month. If the optional HFT attribute is not available, default
 6566 to midnight local time.

6567 D.3.2.2.13.27 PreviousMonthNAlternativeConsumptionReceived Attribute

6568 *PreviousMonthNAlternativeConsumptionReceived* represents the summed value received from
 6569 the premises within the previous month period starting at the *Historical Freeze Time* (HFT) on
 6570 the 1st of the month to the last day of the month. If the optional HFT attribute is not available,
 6571 default to midnight local time.

6572

6573

6574 D.3.2.3 Server Commands

6575 D.3.2.3.1 Commands Generated

6576 The command IDs generated by the Metering server cluster are listed in Table D-47.

6577

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	

6578

6579 D.3.2.3.1.1 Get Profile Response Command6580 **D.3.2.3.1.1.1 Payload Format**6581 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

6582

Figure D-12– Get Profile Response Command Payload

6583 **D.3.2.3.1.1.2 Payload Details**

6584 **EndTime:** 32-bit value (in UTC) representing the end time of the most chronologically
 6585 recent interval being requested. Example: Data collected from 2:00 PM to 3:00 PM would be
 6586 specified as a 3:00 PM interval (end time). It is important to note that the current interval
 6587 accumulating is not included in most recent block but can be retrieved using the
 6588 *CurrentPartialProfileIntervalValue* attribute.

6589 **Status:** Table D-48 lists the valid values returned in the Status field.

6590 **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

6591
 6592 **ProfileIntervalPeriod:** Represents the interval or time frame used to capture metered
 6593 Energy, Gas, and Water consumption for profiling purposes. *ProfileIntervalPeriod* is
 6594 an enumerated field representing the following timeframes listed in Table D-49:

6595 **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

6596
 6597
 6598 **NumberOfPeriodsDelivered:** Represents the number of intervals the device is returning.
 6599 Please note the number of periods returned in the *Get Profile Response* command can be

6600 calculated when the packets are received and can replace the usage of this field. The intent is to
6601 provide this information as a convenience.

6602 **Intervals:** Series of interval data captured using the period specified by the ProfileIntervalPeriod
6603 field. The content of the interval data depends of the type of information requested using the
6604 Channel field in the *Get Profile* Command, and will represent the change in that information
6605 since the previous interval. Data is organized in a reverse chronological order, the most
6606 recent interval is transmitted first and the oldest interval is transmitted last. Invalid intervals
6607 should be marked as 0xFFFFFFF.

6608 **D.3.2.3.1.1.3 When Generated**

6609 This command is generated when the Client command *GetProfile* is received. Please refer to
6610 sub-clause D.3.3.3.1.1.

6611

6612 **D.3.2.3.1.2 Request Mirror Command**

6613 This command is used to request the ESI to mirror Metering Device data.

6614 **D.3.2.3.1.2.1 Payload Details**

6615 There are no fields for this command.

6616 **D.3.2.3.1.2.2 Effect on Receipt**

6617 On receipt of this command, the Server shall send a *RequestMirrorReponse* command (see
6618 sub-clause D.3.3.3.1.2).

6619

6620 **D.3.2.3.1.3 Remove Mirror Command**

6621 This command is used to request the ESI to remove its mirror of Metering Device data. The
6622 device sending the *Remove Mirror* command to the ESI shall send the command to the mirror
6623 endpoint to be removed. Only the device that created the mirror on the ESI or the ESI itself
6624 should be allowed to remove the mirror from the ESI.

6625 **D.3.2.3.1.3.1 Payload Details**

6626 There are no fields for this command.

6627 **D.3.2.3.1.3.2 Effect on Receipt**

6628 On receipt of this command, the Server shall send a *MirrorRemoved* command (see sub-clause
6629 D.3.3.3.1.3).

6630

6631 D.3.2.3.1.4 Request Fast Poll Mode Response Command

6632 **D.3.2.3.1.4.1 Payload Format**

6633 The *Request Fast Poll Mode Response* command payload shall be formatted as illustrated in
6634 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)

6635 **Figure D-13– Request Fast Poll Mode Response Command Payload**

6636 **D.3.2.3.1.4.2 Payload Details**

6637 **Applied Update Period:** The period at which metering data shall be updated. This may be
6638 different than the requested fast poll. If the Request Fast Poll Rate is less than *Fast Poll Update*
6639 *Period* Attribute, it shall use the Fast Poll Update Period Attribute. Otherwise, the Applied
6640 Update Period shall be greater than or equal to the minimum *Fast Poll Update Period* Attribute
6641 and less than or equal to the Requested Fast Poll Rate

6642 **Fast Poll Mode End Time:** UTC time that indicates when the metering server will terminate
6643 fast poll mode and resume updating at the rate specified by *DefaultUpdatePeriod*. For
6644 example, one or more metering clients may request fast poll mode while the metering server is
6645 already in fast poll mode. The intent is that the fast poll mode will not be extended since this
6646 scenario would make it possible to be in fast poll mode longer than 15 minutes.

6647 **D.3.2.3.1.4.3 When Generated**

6648 This command is generated when the client command *Request Fast Poll Mode* is received.

6649 **D.3.2.3.1.4.4 Effect on Receipt**

6650 On receipt of this command, the device may request or receive updates not to exceed the
6651 Applied Update Period until Fast Poll Mode End Time.

6652

6653 D.3.2.3.1.5 ScheduleSnapshotResponse Command

6654 This command is generated in response to a *ScheduleSnapshot* command, and is sent to confirm
6655 whether the requested snapshot schedule has been set up. See section D.3.4.5 for further details.

6656 **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)

6657

Figure D-14– ScheduleSnapshotResponse Command Payload**D.3.2.3.1.5.2 Payload Details**

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.

D.3.2.3.1.5.3 Snapshot Response Payload

The *ScheduleSnapshotResponse* payload may contain several instances of the sub-payload defined in Figure D-15. Each instance is an acknowledgment from the device for a scheduled 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)

Figure D-15– Snapshot Response Payload Sub-Payload

6667

Snapshot Schedule ID (mandatory): The unique ID of the Snapshot schedule; a range of 1-254 is supported (see D.3.3.3.1.5.2 for further details).

Snapshot Schedule Confirmation (mandatory): This provides confirmation for the Snapshot schedule; enumerations are defined in Table D-50.

6672

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

6673

6674 D.3.2.3.1.6 TakeSnapshotResponse Command

6675 This command is generated in response to a *TakeSnapshot* command, and is sent to confirm
 6676 whether the requested snapshot has been accepted and successfully taken. See section D.3.4.5 for
 6677 further details.

6678 **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)

6679 **Figure D-16– TakeSnapshotResponse Command Payload**6680 **D.3.2.3.1.6.2 Payload Details**

6681 **Snapshot ID (mandatory):** Unique identifier allocated by the device creating the snapshot. The
 6682 value contained in this field indicates the *TakeSnapshot* command for which this response is
 6683 generated.

6684 **Snapshot Confirmation (mandatory):** This is the acknowledgment from the device that it can
 6685 support this required type of snapshot. The enumerations are defined in Table D-51.

6686 **Table D-51– Snapshot Confirmation**

Enumeration	Description
0x00	Accepted
0x01	Snapshot Cause not supported
0x02 – 0xFF	Reserved

6687

6688 D.3.2.3.1.7 Publish Snapshot Command

6689 This command is generated in response to a *GetSnapshot* command. It is used to return a single
 6690 snapshot to the client. See section D.3.4.5 for further details.

6691 **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 Payload (M)

6692 **Figure D-17– Publish Snapshot Command Payload**

6693 **D.3.2.3.1.7.2 Payload Details**6694 **Snapshot ID (mandatory):** Unique identifier allocated by the device creating the snapshot.6695 **Snapshot Time (mandatory):** This is a 32 bit value (in UTC Time) representing the time at
6696 which the data snapshot was taken.6697 **Total Snapshots Found (mandatory):** An 8-bit Integer indicating the number of snapshots
6698 found, based on the search criteria defined in the associated *GetSnapshot* command. If the value
6699 is greater than 1, the client is able to request the next snapshot by incrementing the *Snapshot*
6700 *Offset* field in an otherwise repeated *GetSnapshot* command.6701 **Command Index (mandatory):** The *CommandIndex* is used to count the payload fragments in
6702 the case where the entire payload (snapshot) does not fit into one message. The *CommandIndex*
6703 starts at 0 and is incremented for each fragment belonging to the same command.6704 **Total Number of Commands (mandatory):** In the case where the entire payload (snapshot)
6705 does not fit into one message, the *Total Number of Commands* field indicates the total number of
6706 sub-commands that will be returned.6707 **Snapshot Cause (mandatory):** A 32-bit BitMap indicating the cause of the snapshot. The
6708 snapshot cause values are listed in Table D-52.6709 **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

6710 **SnapshotPayloadType (mandatory):** The *SnapshotPayloadType* is an 8-bit enumerator
 6711 defining the format of the *SnapshotPayload* in this message. The different snapshot types are
 6712 listed in Table D-53. The server selects the *SnapshotPayloadType* based on the charging scheme
 6713 in use.

6714

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

6715 If the snapshot is taken by the server due to a change of Tariff Information (cause = 3) which
 6716 involves a change in charging scheme then two snapshots shall be taken, the first according to the
 6717 charging scheme being dismissed, the second to the scheme being introduced.

6718 **SnapshotPayload (mandatory):** the format of the *SnapshotPayload* differs depending on the
 6719 *SnapshotPayloadType*:

6720 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)
---------------------------------	--------------------------	------------------------------------	------------------------------	---------------------------------------	-----------------------------------	----------------------------	--------------------

6721

Figure D-18– TOU Information Delivered Snapshot Payload6722
6723

Current Summation Delivered (mandatory): An unsigned 48-bit integer that returns the value of the *CurrentSummationDelivered* attribute at the stated snapshot timestamp.

6724
6725
6726

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.

6727
6728

BillToDateTimeStampDelivered (mandatory): A UTC timestamp that indicates when the value of the associated *BillToDateDelivered* parameter was last updated.

6729
6730
6731
6732

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.

6733
6734

ProjectedBillTimeStampDelivered (mandatory): A UTC timestamp that indicates when the associated *ProjectedBillDelivered* parameter was last updated.

6735
6736
6737
6738

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.

6739
6740

Number of Tiers in Use (mandatory): An 8-bit integer representing the number of tiers in use at the time the snapshot was taken.

6741
6742
6743
6744
6745
6746
6747

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.

b SnapshotPayloadType = TOU Information Received Set

6	4	4	4	4	1	1	Variable
Unsigned 48-bit Integer	Unsigned 32-bit Integer	UTC Time	Unsigned 32-bit Integer	UTC Time	8-bit BitMap	Unsigned 8-bit Integers	Series of Unsigned 48-bit Integers

Current Summation Received (M)	BillToDate Received (M)	BillToDate TimeStamp Received (M)	Projected Bill Received (M)	ProjectedBill TimeStamp Received (M)	Bill Received Trailing Digit (M)	Number of Tiers in Use (M)	Tier Summation (M)
--------------------------------	-------------------------	-----------------------------------	-----------------------------	--------------------------------------	----------------------------------	----------------------------	--------------------

Figure D-19– TOU Information Received Snapshot Payload

6748 **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.

6752 **BillToDateReceived (mandatory):** An unsigned 32-bit integer that provides a value for the costs in the current billing period. This value is measured in a base unit of Currency with the decimal point located as indicated by the *BillReceivedTrailingDigit* field.

6755 **BillToDateTimeStampReceived (mandatory):** A UTC timestamp that indicates when the value of the associated *BillToDateReceived* parameter was last updated.

6757 **ProjectedBillReceived (mandatory):** An unsigned 32-bit integer that provides a value indicating what the estimated state of the account will be at the end of the billing period based on past generation. This attribute is measured in a base unit of Currency with the decimal point located as indicated by the *BillReceivedTrailingDigit* field.

6761 **ProjectedBillTimeStampReceived (mandatory):** A UTC timestamp that indicates when the associated *ProjectedBillReceived* parameter was last updated.

6763 **BillReceivedTrailingDigit (mandatory):** An 8-bit BitMap used to determine where the decimal point is located in the *BillToDateReceived* and *ProjectedBillReceived* fields. The most significant nibble indicates the number of digits to the right of the decimal point. The least significant nibble is reserved and shall be 0.

6767 **Number of Tiers in Use (mandatory):** An 8-bit integer representing the number of tiers in use at the time the snapshot was taken.

6769 **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.

6774 c) SnapshotPayloadType = Block Information Delivered Set

6	4	4	4	4	1
Unsigned 48-bit Integer	Unsigned 32-bit Integer	UTC Time	Unsigned 32-bit Integer	UTC Time	8-bit BitMap

Current Summation Delivered (M)	BillToDate Delivered (M)	BillToDate TimeStamp Delivered (M)	Projected Bill Delivered (M)	ProjectedBill TimeStamp Delivered (M)	Bill Delivered Trailing Digit (M)
6776	1	Variable	1	Variable	
	Unsigned 8-bit Integers	Series of Unsigned 48-bit Integers	Unsigned 8-bit BitMap	Series of Unsigned 48-bit Integers	
6777	Number of Tiers in Use (M)	Tier Summation (M)	Number of Tiers and Block Thresholds in Use (M)	Tier Block Summation (M)	

Figure D-20—Block Information Delivered SnapshotPayload

Current Summation Delivered (mandatory): An unsigned 48-bit integer that returns the value of the *CurrentSummationDelivered* attribute at the stated snapshot timestamp.

BillToDateDelivered (mandatory): An unsigned 32-bit integer that provides a value for the costs in the current billing period. This value is measured in a base unit of Currency with the decimal point located as indicated by the *BillDeliveredTrailingDigit* field.

BillToDateTimeStampDelivered (mandatory): A UTC timestamp that indicates when the value of the associated *BillToDateDelivered* parameter was last updated.

ProjectedBillDelivered (mandatory): An unsigned 32-bit integer that provides a value indicating what the estimated state of the account will be at the end of the billing period based on past consumption. This attribute is measured in a base unit of Currency with the decimal point located as indicated by the *BillDeliveredTrailingDigit* field.

ProjectedBillTimeStampDelivered (mandatory): A UTC timestamp that indicates when the associated *ProjectedBillDelivered* parameter was last updated.

BillDeliveredTrailingDigit (mandatory): An 8-bit BitMap used to determine where the decimal point is located in the *BillToDateDelivered* and *ProjectedBillDelivered* fields. The most significant nibble indicates the number of digits to the right of the decimal point. The least significant nibble is reserved and shall be 0.

Number of Tiers in Use (mandatory): An 8-bit integer representing the number of tiers in use at the time the snapshot was taken.

TierSummation (mandatory): The *Publish Snapshot* command contains N elements of *CurrentTierNSummationDelivered* attributes from the TOU Information Set. The Metering server shall send only the number of tiers in use, as stated in this command. The first element of the TOU Information Set (Delivered), is *CurrentTier1Summation*. For the following elements, the tier index is incremented until the number of tiers in use is reached.

Number of Tiers and Block Thresholds in Use (mandatory): An 8-bit BitMap representing the number of tiers and block thresholds in use at the time the snapshot was taken. The most significant nibble defines the number of tiers in use, whereas the least significant nibble indicates the number of block thresholds in use.

TierBlockSummation (T,B): The *Publish Snapshot* command contains N elements of the Block Information Attribute Set (Delivered). The metering server shall send only the number of Tiers and Blocks in use as stated in this command. The Block Information Attribute Set has two dimensions, the row – tier index (T) and the block – column index (B).

The first element of the Tier Block Summation field is *CurrentTier1Block1SummationDelivered* attribute. For the following elements, the block index is incremented until the number of blocks in use is reached. Then the tier index is incremented and the block index starts at 1 again. This continues until the stated number of tiers in use is reached.

6815 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)	BillToDateTimeStamp Received (M)	Projected Bill Received (M)	ProjectedBillTimeStamp 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)

6817 **Figure D-21– Block Information Received SnapshotPayload**

Current Summation Received (mandatory): An unsigned 48-bit integer that returns the value of the *CurrentSummationReceived* attribute at the stated snapshot timestamp. A value of 0xFFFFFFFFFFFF means not available.

BillToDateReceived (mandatory): An unsigned 32-bit integer that provides a value for the costs in the current billing period. This value is measured in a base unit of Currency with the decimal point located as indicated by the *BillReceivedTrailingDigit* field.

BillToDateTimeStampReceived (mandatory): A UTC timestamp that indicates when the value of the associated *BillToDateReceived* parameter was last updated.

6826 **ProjectedBillReceived (mandatory):** An unsigned 32-bit integer that provides a value
 6827 indicating what the estimated state of the account will be at the end of the billing period based
 6828 on past generation. This attribute is measured in a base unit of Currency with the decimal
 6829 point located as indicated by the *BillReceivedTrailingDigit* field.

6830 **ProjectedBillTimeStampReceived (mandatory):** A UTC timestamp that indicates when the
 6831 associated *ProjectedBillReceived* parameter was last updated.

6832 **BillReceivedTrailingDigit (mandatory):** An 8-bit BitMap used to determine where the
 6833 decimal point is located in the *BillToDateReceived* and *ProjectedBillReceived* fields. The
 6834 most significant nibble indicates the number of digits to the right of the decimal point. The
 6835 least significant nibble is reserved and shall be 0.

6836 **Number of Tiers in Use (mandatory):** An 8-bit integer representing the number of tiers in
 6837 use at the time the snapshot was taken.

6838 **TierSummation (mandatory):** The *Publish Snapshot* command contains N elements of
 6839 *CurrentTierNSummationReceived* attributes from the TOU Information Set. The Metering
 6840 server shall send only the number of tiers in use, as stated in this command. The first element
 6841 of the TOU Information Set (Received), is *CurrentTier1Summation*. For the following
 6842 elements, the tier index is incremented until the number of tiers in use is reached.

6843 **Number of Tiers and Block Thresholds in Use (mandatory):** An 8-bit BitMap representing
 6844 the number of tiers and block thresholds in use at the time the snapshot was taken. The most
 6845 significant nibble defines the number of tiers in use, whereas the least significant nibble
 6846 indicates the number of block thresholds in use.

6847 **TierBlockSummation (T,B):** The *Publish Snapshot* command contains N elements of the
 6848 Block Information Attribute Set (Received). The metering server shall send only the number
 6849 of Tiers and Blocks in use as stated in this command. The Block Information Attribute Set has
 6850 two dimensions, the row – tier index (T) and the block – column index (B).

6851 The first element of the Tier Block Summation field is
 6852 *CurrentTier1Block1SummationReceived* attribute. For the following elements, the block index
 6853 is incremented until the number of blocks in use is reached. Then the tier index is incremented
 6854 and the block index starts at 1 again. This continues until the stated number of tiers in use is
 6855 reached.

6856 e [SnapshotPayloadType = TOU Information Set Delivered \(No Billing\)](#)

6	1	Variable
Unsigned 48-bit Integer	Unsigned 8-bit Integers	Series of Unsigned 48-bit Integers

Current Summation Delivered (M)	Number of Tiers in Use (M)	Tier Summation (M)
---------------------------------	----------------------------	--------------------

Figure D-22– TOU Information Delivered (No Billing) Snapshot 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.

6868 f SnapshotPayloadType = TOU Information Set Received (No Billing)

6	1	Variable
Unsigned 48-bit Integer	Unsigned 8-bit Integers	Series of Unsigned 48-bit Integers
Current Summation Received (M)	Number of Tiers in Use (M)	Tier Summation (M)

Figure D-23– TOU Information Received (No Billing) Snapshot 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.

6881 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)

6882 **Figure D-24—Block Information Delivered (No Billing) Snapshot Payload**

6883 **Current Summation Delivered (mandatory):** An unsigned 48-bit integer that returns the
 6884 value of the *CurrentSummationDelivered* attribute at the stated snapshot timestamp.

6885 **Number of Tiers in Use (mandatory):** An 8-bit integer representing the number of tiers in
 6886 use at the time the snapshot was taken.

6887 **TierSummation (mandatory):** The *Publish Snapshot* command contains N elements of
 6888 *CurrentTierNSummationDelivered* attributes from the TOU Information Set. The Metering
 6889 server shall send only the number of tiers in use, as stated in this command. The first element
 6890 of the TOU Information Set (Delivered), is *CurrentTier1Summation*. For the following
 6891 elements, the tier index is incremented until the number of tiers in use is reached.

6892 **Number of Tiers and Block Thresholds in Use (mandatory):** An 8-bit BitMap representing
 6893 the number of tiers and block thresholds in use at the time the snapshot was taken. The most
 6894 significant nibble defines the number of tiers in use, whereas the least significant nibble
 6895 indicates the number of block thresholds in use.

6896 **TierBlockSummation (T,B):** The *Publish Snapshot* command contains N elements of the
 6897 Block Information Attribute Set (Delivered). The metering server shall send only the number
 6898 of Tiers and Blocks in use as stated in this command. The Block Information Attribute Set has
 6899 two dimensions, the row – tier index (T) and the block – column index (B).

6900 The first element of the Tier Block Summation field is
 6901 *CurrentTier1Block1SummationDelivered* attribute. For the following elements, the block
 6902 index is incremented until the number of blocks in use is reached. Then the tier index is
 6903 incremented and the block index starts at 1 again. This continues until the stated number of
 6904 tiers in use is reached.

6905 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)
--------------------------------	----------------------------	--------------------	---	--------------------------

6906

Figure D-25– Block Information Received (No Billing) Snapshot Payload

6907 **Current Summation Received (mandatory):** An unsigned 48-bit integer that returns the
 6908 value of the *CurrentSummationReceived* attribute at the stated snapshot timestamp. A value of
 6909 0xFFFFFFFFFFFF means not available.

6910 **Number of Tiers in Use (mandatory):** An 8-bit integer representing the number of tiers in
 6911 use at the time the snapshot was taken.

6912 **TierSummation (mandatory):** The *Publish Snapshot* command contains N elements of
 6913 *CurrentTierNSummationReceived* attributes from the TOU Information Set. The Metering
 6914 server shall send only the number of tiers in use, as stated in this command. The first element
 6915 of the TOU Information Set (Received), is *CurrentTier1Summation*. For the following
 6916 elements, the tier index is incremented until the number of tiers in use is reached.

6917 **Number of Tiers and Block Thresholds in Use (mandatory):** An 8-bit BitMap representing
 6918 the number of tiers and block thresholds in use at the time the snapshot was taken. The most
 6919 significant nibble defines the number of tiers in use, whereas the least significant nibble
 6920 indicates the number of block thresholds in use.

6921 **TierBlockSummation (T,B):** The *Publish Snapshot* command contains N elements of the
 6922 Block Information Attribute Set (Received). The metering server shall send only the number
 6923 of Tiers and Blocks in use as stated in this command. The Block Information Attribute Set has
 6924 two dimensions, the row – tier index (T) and the block – column index (B).

6925 The first element of the Tier Block Summation field is
 6926 *CurrentTier1Block1SummationReceived* attribute. For the following elements, the block index
 6927 is incremented until the number of blocks in use is reached. Then the tier index is incremented
 6928 and the block index starts at 1 again. This continues until the stated number of tiers in use is
 6929 reached.

6930 D.3.2.3.1.7.3 When Generated

6931 A *Publish Snapshot* command is generated in response to *GetSnapshot* command. The device
 6932 shall send a single *Publish Snapshot* command according to the search criteria defined in the
 6933 associated *GetSnapshot* command. A ZCL Default Response with status NOT_FOUND shall be
 6934 returned if there is no appropriate snapshot data available.

6935 D.3.2.3.1.8 GetSampledDataResponse Command

6936 This command is used to send the requested sample data to the client. It is generated in response
 6937 to a *GetSampledData* command (see D.3.3.3.1.9).

6938

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)	SampleRequestInterval (M)	NumberOfSamples (M)	Samples (M)

6939

Figure D-26– Format of the *GetSampledDataResponse* Command Payload

6940

D.3.2.3.1.8.2 Payload Details6941
6942

SampleID (mandatory): Unique identifier allocated to this Sampling session. This field allows devices to match response data with the appropriate request. See D.3.2.3.1.14 for further details.

6943
6944

SampleStartTime (mandatory): A UTC Time field to denote the time of the first sample returned in this response.

6945
6946

SampleType (mandatory): An 8 bit enumeration that identifies the type of data being sampled. Possible values are defined in the following table:

6947

Table D-54– Sample Type Enumerations

Enumeration	Description
0	Consumption Delivered
1 - 255	Reserved

6948

SampleRequestInterval (mandatory): An unsigned 16-bit field representing the interval or time in seconds between samples.

6949
6950
6951
6952
6953
6954
6955

NumberOfSamples (mandatory): Represents the number of samples being requested, This value cannot exceed the size stipulated in the MaxNumberofSamples field in the *StartSampling* command. If more samples are requested than can be delivered, the *GetSampleDataResponse* command will return the number of samples equal to MaxNumberofSamples field. If fewer samples are available for the time period, only those available shall be returned.

6956
6957
6958
6959
6960

Samples (mandatory): Series of data samples captured using the interval specified by the *SampleRequestInterval* field in the *StartSampling* command. Each sample contains the change in the relevant data since the previous sample. Data is organised in a chronological order, the oldest sample is transmitted first and the most recent sample is transmitted last. Invalid samples should be marked as 0xFFFF.

6961 **D.3.2.3.1.8.3 When Generated**

6962 A *GetSampledDataResponse* command is generated in response to *GetSampledData* command.
 6963 A ZCL Default Response with status NOT_FOUND shall be returned if there is no appropriate
 6964 Sample data available.

6965

6966

6967 **D.3.2.3.1.9 ConfigureMirror Command**

6968 Where ‘Two Way Mirroring’ is being implemented, this command shall be sent to the mirror
 6969 once the mirror has been created. The command allows a BOMD to provide the operational
 6970 configuration of the associated Mirror. Note that this command is not required for a traditional
 6971 ‘One way’ mirror (see D.3.4.4.3 for further details).

6972 **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)

6973 **Figure D-27– Format of the *ConfigureMirror* Command Payload**6974 **D.3.2.3.1.9.2 Payload Details**

6975 **Issuer Event ID (mandatory):** Unique identifier generated by the device being mirrored. When
 6976 new information is provided that replaces older information, this field allows devices to
 6977 determine which information is newer. It is recommended that the value contained in this field is
 6978 a UTC based time stamp (UTCTime data type) identifying when the command was issued. Thus,
 6979 newer information will have a value in the Issuer Event ID field that is larger than older
 6980 information.

6981 **Reporting Interval (mandatory):** An unsigned 24-bit integer to denote the interval, in seconds,
 6982 at which a mirrored meter should use the *ReportAttribute* command..

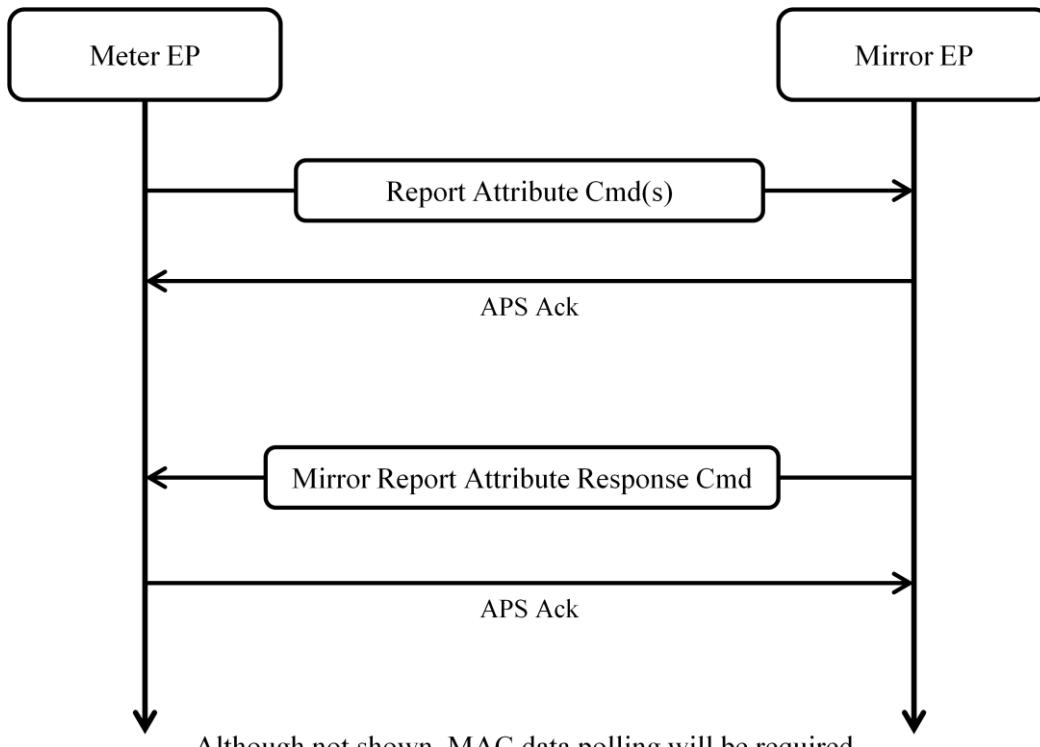
6983 **Mirror Notification Reporting (mandatory):** A Boolean used to advise a BOMD how the
 6984 Notification flags should be acquired (see below).

6985 **Notification Scheme (mandatory):** see section D.3.3.2.1.3.

6986 When *Mirror Notification Reporting* is set, the *MirrorReportAttributeResponse* command is
 6987 enabled. In that case, the Metering client on the mirror endpoint shall respond to the last or only

6988 ReportAttribute command with the *MirrorReportAttributeResponse*. This is shown in Figure
 6989 D-28:

6990



6991

6992

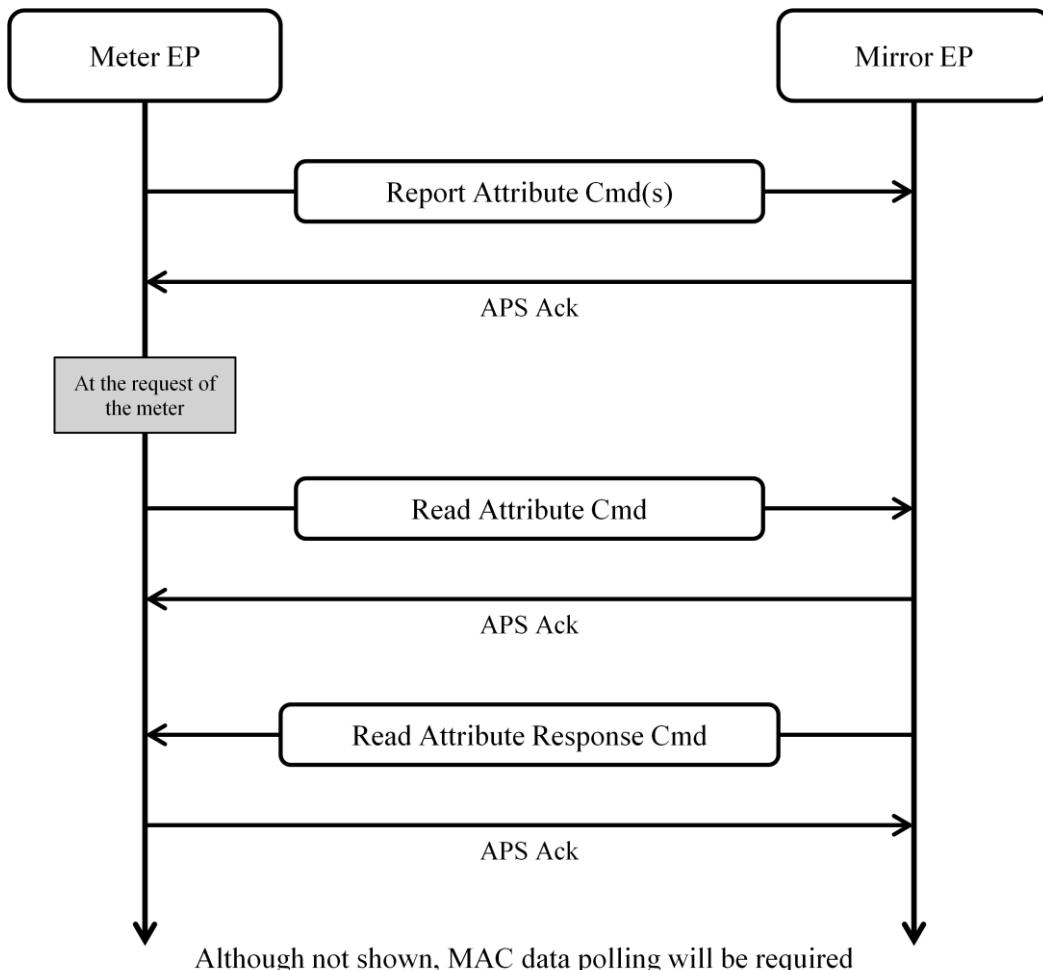
Figure D-28—*MirrorReportAttributeResponse* Command Enabled

6993 **NOTES:**

- 6994 1. On powering up, the BOMD will send one or more *Report Attribute* commands to the
 6995 Metering client on the mirror endpoint. The last attribute to be reported to the mirror shall be
 6996 an *Attribute Reporting Status* attribute, as defined in section A.2.
- 6997 2. If *MirrorReportAttributeResponse* is enabled, the server does not need to request an APS
 6998 ACK. If the server requests an APS ACK, the Metering client on the mirror endpoint shall
 6999 respond first with an APS ACK and then send the *MirrorReportAttributeResponse*.

7000 If the *MirrorReportAttributeResponse* command is not enabled, the Metering server may poll the
 7001 Notification flags by means of a normal *ReadAttribute* command, as shown in Figure D-29:

7002



7003

7004

7005

7006 **D.3.2.3.1.9.3 When Generated**7007 The *ConfigureMirror* command is generated in response to the *RequestMirrorResponse* command when the Mirror has been created.7009 **D.3.2.3.1.9.4 Effect on Receipt**7010 On receipt of the *ConfigureMirror* command, the mirror will understand if the 7011 *MirrorReportAttributeResponse* command should be sent, and if there is a scheme for the 7012 Notifications flags. The Mirror will also understand the interval at which the Meter shall report to 7013 the mirror.

7014 A ZCL Default Response with status INVALID_FIELD shall be returned if the required 7015 Notification Scheme is not supported by the Mirroring device.

7016

7017 D.3.2.3.1.10 ConfigureNotificationScheme Command

7018 Where ‘Two Way Mirroring’ is being implemented, and a non-default Notification Scheme is to
 7019 be used, this command shall be sent to the mirror once the mirror has been created. The
 7020 command allows a BOMD to provide details of the required Notification Scheme to the
 7021 associated mirror, and should be used in conjunction with the associated
 7022 *ConfigureNotificationFlags* command (see D.3.2.3.1.11). No default schemes are allowed to be
 7023 overwritten (see D.3.4.4.3.2 and D.3.4.4.3.3 for further details).

7024 **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)

7025 **Figure D-30—ConfigureNotificationScheme Command Payload**7026 **D.3.2.3.1.10.2 Payload Details**

7027 **Issuer Event ID (mandatory):** Unique identifier generated by the device being mirrored. When
 7028 new information is provided that replaces older information, this field allows devices to
 7029 determine which information is newer. It is recommended that the value contained in this field is
 7030 a UTC based time stamp (UTCTime data type) identifying when the command was issued. Thus,
 7031 newer information will have a value in the Issuer Event ID field that is larger than older
 7032 information.

7033 **Notification Scheme (mandatory):** see section D.3.3.2.1.3

7034 **Notification Flag Order (mandatory):** A 32-bit bitmap, consisting of 8 nibbles which define
 7035 the Notification Flag attributes (and order) to be returned in a *MirrorReportAttributeResponse*
 7036 command. The values to be returned in each nibble are defined in Table D-55.

7037

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

7038 **D.3.2.3.1.10.3 When Generated**

7039 The *ConfigureNotificationScheme* command is generated when a new scheme is required.

7040 **D.3.2.3.1.10.4 Effect on Receipt**

7041 On receipt of the *ConfigureNotificationScheme* command, the mirror shall update the
 7042 *NotificationScheme* attribute, and wait for the *ConfigureNotificationFlags* commands. Until all
 7043 of the *ConfigureNotificationFlags* commands have been received, the two-way mirror
 7044 functionality should be disabled. The *Notification Flag Order* parameter will allow the mirror to
 7045 determine when all of the *ConfigureNotificationFlags* commands have been received.

7046

7047 D.3.2.3.1.11 ConfigureNotificationFlags Command

7048 Where ‘Two Way Mirroring’ is being implemented, and a non-default *Notification Scheme* is to
 7049 be used, the *ConfigureNotificationFlags* command allows a BOMD to set the commands relating
 7050 to the bit value for each *NotificationFlags#N* attribute that the scheme is proposing to use. This
 7051 command should be used in conjunction with the associated *ConfigureNotificationScheme*
 7052 command (see D.3.2.3.1.10). No predefined schemes are allowed to be overwritten (see
 7053 D.3.4.4.3.2 and D.3.4.4.3.3 for further details).

7054 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

7055 Figure D-31– *ConfigureNotificationFlags* Command Payload

7056 D.3.2.3.1.11.2 *Payload Details*

7057 **Issuer Event ID (mandatory):** Unique identifier generated by the device being mirrored. When
 7058 new information is provided that replaces older information, this field allows devices to
 7059 determine which information is newer. It is recommended that the value contained in this field is
 7060 a UTC based time stamp (UTCTime data type) identifying when the command was issued. Thus,

7061 newer information will have a value in the Issuer Event ID field that is larger than older
 7062 information.

7063 **Notification Scheme (mandatory):** see section D.3.3.2.1.3

7064 **Notification Flag Attribute ID (mandatory):** An unsigned 16-bit integer that denotes the
 7065 attribute id of the Notification flag (2-8) that will be configured for this Notification scheme.

7066 **Bit Field Allocation (mandatory):** The bit field allocation sub payload is defined in Figure
 7067 D-32. 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

7068 **Figure D-32– Bit Field Allocation Command Sub Payload**

7069 **Cluster ID (mandatory):** An unsigned 16-bit integer that denotes the Cluster id of the
 7070 Notification flag that will be configured for this Notification scheme.

7071 **Manufacturer Code (mandatory):** An unsigned 16-bit integer that denotes the Manufacturer
 7072 Code to be used with these command IDs, that are configured for this Notification flag within
 7073 this Notification scheme.

7074 **No of Commands (mandatory):** An unsigned 8-bit integer that indicates the number of
 7075 command identifiers contained within this sub payload.

7076 **Command ID (mandatory):** An unsigned 8-bit integer that denotes the command that is to be
 7077 used. The command id should be used with the cluster id to reference the command(s).

7078 **D.3.2.3.1.11.3 When Generated**

7079 This command is sent once the mirror has been created, and the *ConfigureNotificationScheme*
 7080 command has been sent up the top level of the scheme. There is a *ConfigureNotificationFlags*
 7081 command for each attribute that the scheme is proposing to use. No default schemes are allowed
 7082 to be overwritten.

7083 **D.3.2.3.1.11.4 Effect on Receipt**

7084 Once all *ConfigureNotificationFlags* commands have been received, a fully populated scheme
 7085 will be available, and two way mirroring can then be enabled.

7086

7087 D.3.2.3.1.12 GetNotifiedMessage Command

7088 The *GetNotifiedMessage* command is used only when a BOMD is being mirrored. This
7089 command provides a method for the BOMD to notify the Mirror message queue that it wants to
7090 receive commands that the Mirror has queued. The Notification flags set within the command
7091 shall inform the mirror of the commands that the BOMD is requesting.

7092 **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)

7093 **Figure D-33– *GetNotifiedMessage* Command Payload**

7094 **D.3.2.3.1.12.2 Payload Details**

7095 **Notification Scheme (mandatory):** see section D.3.3.2.1.3.

7096 **Notification Flag Attribute ID (mandatory):** An unsigned 16-bit integer that denotes the
7097 attribute id of the notification flag (1-8) that is included in this command.

7098 **Notification Flags #N (mandatory):** The Notification Flags attribute/parameter indicating the
7099 command being requested. See D.3.3.2.1.1 and D.3.3.2.1.2 for further details.

7100 **D.3.2.3.1.12.3 When Generated**

7101 The *GetNotifiedMessage* command is generated in response to the flags that have been set within
7102 the *NotificationFlags#N* attribute/parameter within the *MirrorReportAttributeResponse*
7103 command. The BOMD shall be in control of when it sends this command and what commands it
7104 shall request. This command should only be generated when there is no specific “GET”
7105 command to be used to fetch the information. i.e. if the scheme supports *GetProfile* &
7106 *GetProfileResponse*, the attribute could be configured to inform the BOMD that the mirror
7107 requires some load profile information. Therefore, by setting the flag in this command, the
7108 BOMD is requesting that the *GetProfile* command is now sent to it.

7109 The BOMD may choose not to initiate the process if the battery level does not allow it, or if the
7110 request is sent too often.

7111 **D.3.2.3.1.12.4 Effect on Receipt**

7112 Dependent on the flags set within the command, the Mirror shall send down the appropriate
7113 command to the BOMD.

7114

7115 D.3.2.3.1.13 Supply Status Response Command

7116 This command is transmitted by a Metering Device in response to a *Change Supply* command.

7117 **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)

7118 **Figure D-34– Supply Status Response Command Payload**

7119 **D.3.2.3.1.13.2 Payload Details**

7120 **Provider ID (mandatory):** An unsigned 32-bit field containing a unique identifier for the commodity provider to whom this command relates.

7122 **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.

7128 **Implementation Date/Time (mandatory):** A UTC Time field to indicate the date at which the originating command was to be applied.

7130 **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.

7133 **Table D-56– Supply Status Field Enumerations**

Enumerated Value	Status
0x00	Supply OFF
0x01	Supply OFF / ARMED
0x02	Supply ON

7134

7135 **D.3.2.3.1.13.3 When Generated**

7136 This command is transmitted by a Metering Device to indicate that a *Change Supply* command has been successfully executed. It shall be sent if an acknowledgment is requested in the originating command (see sub-clause D.3.3.3.1.12).

7139

7140 D.3.2.3.1.14 Start Sampling Response Command7141 This command is transmitted by a Metering Device in response to a *StartSampling* command.7142 **D.3.2.3.1.14.1 Payload Format**7143 The *StartSamplingResponse* command payload shall be formatted as illustrated in Figure D-38

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

7144 **Figure D-35– *StartSamplingResponse* Command Payload**7145 **D.3.2.3.1.14.2 Payload Details**7146 **Sample ID:** 16 Bit Unsigned Integer indicating the ID allocated by the Metering Device for the requested Sampling session. If the Metering Device is unable to support a further Sampling session, *Sample ID* shall be returned as 0xFFFF. If valid, the *Sample ID* shall be used for all further communication regarding this Sampling session.7150 NOTE that the Metering Device may reserve a *Sample ID* of 0x0000 in order to provide an alternative mechanism for retrieving Profile data. This mechanism will allow an increased number of samples to be returned than is available via the existing (automatically started) Profile mechanism.

7154

7155 **D.3.3 Client**7156 **D.3.3.1 Dependencies**

7157 No additional dependencies.

7158 **D.3.3.2 Attributes**

7159 For convenience, the attributes defined in this specification are arranged into sets of related attributes; each set can contain up to 256 attributes. Attribute identifiers are encoded such that the most significant Octet specifies the attribute set and the least significant Octet specifies the attribute within the set. The currently defined attribute sets are listed in Table D-57.

7163

Table D-57 – Metering Cluster Client Attribute Sets

Attribute Set Identifier	Description
0x00	Notification Attribute Set

0x01 to 0xFF	Reserved
--------------	----------

7164

7165 D.3.3.2.1 **Notification Attribute Set**

7166 The Notification Attribute Set is used to notify battery operated mirrored devices (BOMDs) that
7167 the ESI or other HAN device has pending information which should be fetched.

7168 Only clients on a mirror endpoint shall support this attribute set.

7169 When commands / attributes are received into the ESI from the HES or other HAN devices, the
7170 ESI will store the corresponding information and set the appropriate bits in the Notification Flag
7171 attributes (BitMaps). The ESI shall reset the bit as soon as a ‘Get’ command with the
7172 corresponding message type is received and all commands of the appropriate type have been
7173 retrieved (this is to allow for multiple commands of the same type).

7174

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
0x0007	NotificationFlags8	32 bit BitMap	0x00000000 - 0xFFFFFFFF	Read	0	O
0x0008 - 0x00FF	Reserved					

7175

7176 D.3.3.2.1.1 FunctionalNotificationFlags

7177 The *FunctionalNotificationFlags* attribute is implemented as a set of bit flags which are have a
 7178 predefined action associated with a bit that is not based on a specific command, but may require
 7179 the Mirrored device to trigger some additional functionality within the system. The Bit Flags are
 7180 defined as shown below:

7181 **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
22-31	Reserved

7182
 7183 **New OTA Firmware Flag:** will be set by the ESI, when the ESI has new OTA Firmware to send
 7184 to the BOMD. The BOMD can then make the decision on when it starts the OTA upgrade
 7185 request.

7186 **CBKE Update Request Flag:** requests the BOMD to initiate the CBKE process with the Trust
 7187 Center to replace the link key currently in use.

7188 **Time Sync Request Flag:** requests the BOMD to initiate the time synchronization process with
 7189 the Time server.

7190 **The Stay Awake Request Flags:** will be set by the ESI when the ESI wants to send a command.
 7191 There are two types of the Stay Awake requests, one for HAN requests and one for WAN
 7192 requests; an implementation may react differently depending on the source of the request:

7193 **HAN requests:** The HAN Stay Awake flag should only be used for commands that
 7194 originate from HAN and not from the ESI that is supporting the commodity of the
 7195 Mirrored device.

7196 **WAN requests:** The WAN Stay Awake flag should only be used for commands that
 7197 originate from the backhaul network.

7198 **Table D-60 – Example Usage of Stay Awake Request Flags**

Waiting Command
Get Snapshot
Schedule Snapshot
Take Snapshot
Start Logging
Get Logging
Get Profile
GetEventLog
ClearEventLog
Reset Demand limit Counter
Read Attribute
Write Attribute

7199 Table D-60 shows example usage of the Stay Awake Request flags. The most likely use is
 7200 when profile data or snapshots are required. The commands shown require additional
 7201 parameters to be sent by the requesting device to solicit the correct response from the
 7202 receiver.

7204 A battery operated meter should read the Notification Flags regularly or enable the
 7205 *MirrorReportAttributeResponse* command. If the *StayAwakeRequest* flag is set, the
 7206 battery operated meter shall poll its parent node at least three times for pending messages.
 7207 The polling interval shall be configurable and not less than 250 ms.

7208 The ESI shall try to send commands after the BOMD has pushed meter readings or read
 7209 the Notification Flags. After reception of a command, the BOMD shall read the
 7210 *NotificationFlags* again. If the ESI has successfully transmitted all pending commands, it
 7211 shall reset the *StayAwakeRequest* flag.

7212 Nevertheless, the BOMD can decide to go to sleep if the *StayAwakeRequest* flag is not
 7213 reset after consecutive reads of the *NotificationFlags* attribute or if it is required by its
 7214 power supply constraints.

7215 **Push Historical Metering Data Attribute Set:** This notification flag requests the BOMD to
 7216 push a sub set of the historical consumption information found within the Metering cluster's
 7217 'Historical Consumption Attribute Set'. The format of the bits is defined within Table D-61.

7218

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

7219

7220 **Push Historical Payment Data attribute Set:** This notification flag requests the BOMD to push
 7221 a sub set of the historical consumption cost information found within the Prepayment cluster’s
 7222 ‘Historical Cost Consumption Attribute Set’. The format of the bits is defined within Table D-62.

7223

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

7224

7225 **Push All Static Data - Basic Cluster:** This notification flag requests the BOMD to push all of
 7226 the attributes within the Basic cluster that are supported by the mirrored meter.

7227 **Push all static Data - Metering Cluster:** This notification flag requests the BOMD to push all
 7228 of the attributes within the Metering cluster that are supported by the mirrored meter.

7229 **Push All Static Data - Prepayment Cluster:** This notification flag requests the BOMD to push
 7230 all of the attributes within the Prepayment cluster that are supported by the mirrored meter.

7231 **Network Key Active:** When this notification flag has been set, the meter shall check with the TC
 7232 to update the network key.

7233 **Display Message:** When this notification flag has been set, the meter shall send a *Get Last*
 7234 *Message* command to the associated Messaging cluster server (see D.5.3.3.1 for further details).

7235 **Cancel All Messages:** When this notification flag has been set, the meter shall send a *GetMessageCancellation*
 7236 command to the associated Messaging cluster server (see D.5.3.3.3 for
 7237 further details).

7238 **Change Supply Message:** When this notification flag has been set, the meter shall send a
7239 *GetNotifiedMessage* command to the mirror.

7240 **Local Change Supply Message:** When this notification flag has been set, the meter shall send a
7241 *GetNotifiedMessage* command to the mirror.

7242 **SetUncontrolledFlowThreshold** Message: When this notification flag has been set, the meter
7243 shall send a *GetNotifiedMessage* command to the mirror.

Tunnel Message Pending: When set, this notification flag indicates to the BOMD that a message is pending retrieval via the tunnel. If any message(s) is/are pending, then the flag shall be cleared when the last pending message is retrieved.

7247

7248 D.3.3.2.1.2 NotificationFlags Attributes

7249 NotificationFlags2 to NotificationFlags8 are 32-bit bitmaps that each represent a series of flags.
7250 Each flag represents an outstanding command that the Mirror is holding on behalf of the BOMD.
7251 Each flag represents a different command. The format of these attributes is dictated by the
7252 scheme that is currently in operation.

7253

7254

7255 D.3.3.2.1.3 NotificationScheme Attribute

The *NotificationScheme* attribute allows for the pre-loading of the Notification Flags bit mapping to ZCL or Smart Energy Standard commands. The following schemes are currently supported within the Smart Energy Standard:-

Value	Description
0x00	No Notification Scheme Defined
0x01	Predefined Notification Scheme A
0x02	Predefined Notification Scheme B
0x03 – 0x80	Reserved
0x81 – 0xFE	For MSP Requirements
0xFF	Reserved

Figure D-36—NotificationScheme Enumerations

7259

7261 D.3.3.3 Client Commands

7262 D.3.3.3.1 Commands Generated

7263 The command IDs generated by the Metering client cluster are listed in Table D-63.

7264

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	

7265

D.3.3.3.1.1 Get Profile Command7267 The *Get Profile* command payload shall be formatted as illustrated in Figure D-37.

7268

Octets	1	4	1
Data Type	8-bit Enumeration ^a	UTCTime	Unsigned 8-bit Integer
Field Name	Interval Channel	End Time	NumberOfPeriods

7269 a. CCB 1077

7270 **Figure D-37– Get Profile Command Payload**

7271 **D.3.3.3.1.1.1 Payload Details**

7272 **Interval Channel:** Enumerated value used to select the quantity of interest returned by the
 7273 *GetProfileReponse* command. The Interval Channel values are listed in Table D-64.

7274 **Table D-64– Interval Channel Values**

Enumerated Value	Description
0	Consumption Delivered
1	Consumption Received
2 to 255	Not used

7275
 7276 **EndTime:** 32-bit value (in UTCTime) used to select an Intervals block from all the Intervals
 7277 blocks available. The Intervals block returned is the most recent block with its EndTime
 7278 equal or older to the one provided. The most recent Intervals block is requested using an
 7279 End Time set to 0x00000000, subsequent Intervals block are requested using an End time set to
 7280 the EndTime of the previous block - (number of intervals of the previous block *
 7281 ProfileIntervalPeriod).

7282 **NumberofPeriods:** Represents the number of intervals being requested. This value cannot
 7283 exceed the size stipulated in the *MaxNumberOfPeriodsDelivered* attribute. If more intervals are
 7284 requested than can be delivered, the *GetProfileResponse* will return the number of intervals equal
 7285 to *MaxNumberOfPeriodsDelivered*. If fewer intervals are available for the time period, only
 7286 those available are returned.

7287 **D.3.3.3.1.1.2 When Generated**

7288 The *GetProfile* command is generated when a client device wishes to retrieve a list of captured
 7289 Energy, Gas or water consumption for profiling purposes. Due to the potentially large amount
 7290 of profile data available, the client device should store previously gathered data and only
 7291 request the most current data. When initially gathering significant amounts of historical
 7292 interval data, the *GetProfile* command should not be issued any more frequently than 7.5
 7293 seconds to prevent overwhelming the ZigBee network.

7294 **D.3.3.3.1.1.3 Command Processing Response**

7295 If failure occurs in recognizing or processing the payload of the *GetProfile* command, the
 7296 appropriate enumerated ZCL status (as referenced in the ZCL Cluster Library specification)
 7297 will be returned. On success, a non-Default Response is returned without a ZCL status code.

7298 **D.3.3.3.1.1.4 Effect on Receipt**

7299 On receipt of this command, the device shall send a *GetProfileReponse* command (see sub-clause
 7300 D.3.2.3.1.1).

7301

7302 D.3.3.3.1.2 Request Mirror Response Command

7303 The *Request Mirror Response* Command allows the ESI to inform a sleepy Metering Device it
7304 has the ability to store and mirror its data.

7305 **D.3.3.3.1.2.1 Payload Format**

7306 The *Request Mirror Response* command payload shall be formatted as illustrated in Figure D-38

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

7307 **Figure D-38—Request Mirror Response Command Payload**

7308 **D.3.3.3.1.2.2 Payload Details**

7309 **EndPoint ID:** 16 Bit Unsigned Integer indicating the End Point ID to contain the Metering
7310 Devices meter data. Valid End Point ID values are 0x0001 to 0x00F0. If the ESI is able to mirror
7311 the Metering Device data, the low byte of the unsigned 16 bit integer shall be used to contain the
7312 eight bit EndPoint ID. If the ESI is unable to mirror the Metering Device data, EndPoint ID
7313 shall be returned as 0xFFFF. All other EndPoint ID values are reserved. If valid, the Metering
7314 device shall use the EndPoint ID to forward its metered data.

7315

7316 D.3.3.3.1.3 Mirror Removed Command

7317 The *Mirror Removed* Command allows the ESI to inform a sleepy Metering Device mirroring
7318 support has been removed or halted.

7319 **D.3.3.3.1.3.1 Payload Format**

7320 The *Mirror Removed* command payload shall be formatted as illustrated in Figure D-39:

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

7321 **Figure D-39—Mirror Removed Command Payload**

7322 **D.3.3.3.1.3.2 Payload Details**

7323 **Removed EndPoint ID:** 16 Bit Unsigned Integer indicating the End Point ID previously
7324 containing the Metering Devices meter data.

7325

7326 D.3.3.3.1.4 Request Fast Poll Mode Command7327 **D.3.3.3.1.4.1 Payload Format**7328 The *Request Fast Poll Mode* shall be formatted as illustrated in Figure D-40:

Octets	1	1
Data Type	Unsigned 8-bit Integer	Unsigned 8-bit Integer
Field Name	Fast Poll Update Period (seconds)	Duration (minutes)

7329 **Figure D-40– Request Fast Poll Mode Command Payload**7330 **D.3.3.3.1.4.2 Payload Details**7331 **Fast Poll Update Period:** Desired fast poll period not to be less than the
7332 *FastPollUpdatePeriod* attribute.7333 **Duration:** Desired duration for the server to remain in fast poll mode not to exceed 15
7334 minutes as specified in sub-clause D.3.4.2.7335 **D.3.3.3.1.4.3 When Generated**7336 The *Request Fast Poll Mode* command is generated when the metering client wishes to
7337 receive near real-time updates of *InstantaneousDemand*. Fast poll mode shall only be requested
7338 as a result of user interaction (for example, the pushing of a button or activation of fast poll mode
7339 by a menu choice).7340 **D.3.3.3.1.4.4 Effect on Receipt**7341 The metering device may continuously update *InstantaneousDemand* as measurements
7342 are acquired, but at a minimum *InstantaneousDemand* must be updated at the end of each
7343 *FastPollUpdatePeriod*.7344 **D.3.3.3.1.5 ScheduleSnapshot Command**7345 This command is used to set up a schedule of when the device shall create snapshot data. See
7346 section D.3.4.5 for further details.7347 **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)
------------	------------------------	----------------------	---------------------------------	----------------------------------

7348

Figure D-41– ScheduleSnapshot Command Payload

7349

D.3.3.3.1.5.2 Payload Details7350
7351
7352
7353
7354
7355
7356
7357

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 snapshot command was issued. Thus, newer information will have a value in the Issuer Event ID field that is larger than older information. This is required when the snapshot data needs to be transmitted over several messages, allowing for the client to easily identify the set of messages that form a group.

7358
7359
7360

Command Index (mandatory): The *CommandIndex* is used to count the payload fragments for 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.

7361
7362
7363

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.

7364

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 Date (M)	Snapshot Schedule (M)	Snapshot Payload Type (M)	Snapshot Cause (M)

7365

Figure D-42– SnapshotSchedulePayload Format7366
7367
7368

Snapshot Schedule ID (mandatory): The unique ID of the Snapshot schedule; a range of 1-254 is supported, denoting a maximum of 254 different schedules that could be set up within the device.

7369
7370

Snapshot Start Date (mandatory): The *Snapshot Start Date* denotes the date when the Snapshot schedule is to start.

7371
7372
7373

Snapshot Schedule (mandatory): A 24-bit value indicating the schedule that should be used for the snapshot. The snapshot schedule bit field is formatted as indicated in Table D-65.

7374

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

7375 **SnapshotPayloadType (mandatory):** The *SnapshotPayloadType* is an 8-bit enumerator
 7376 defining the format of the *SnapshotPayload* required. The different snapshot types are
 7377 listed in Table D-53. The server selects the *SnapshotPayloadType* based on the charging
 7378 scheme in use.

7379 **Snapshot Cause (mandatory):** A 32-bit BitMap indicating the cause of the snapshot.
 7380 The snapshot cause values are listed in Table D-52.

7381

7382 D.3.3.3.1.6 [TakeSnapshot Command](#)
 7383 This command is used to instruct the cluster server to take a single snapshot. See section D.3.4.5
 7384 for further details.

7385 **D.3.3.3.1.6.1 Payload Format**

Octets	4
Data Type	32-bit BitMap
Field Name	Snapshot Cause (M)

7386 Figure D-43– *TakeSnapshot Command Payload*

7387 **D.3.3.3.1.6.2 Payload Details**

7388 **Snapshot Cause (mandatory):** A 32-bit BitMap indicating the cause of the snapshot. The
 7389 snapshot cause values are listed in Table D-52. Note that the *Manually Triggered from Client*
 7390 flag shall additionally be set for all Snapshots triggered in this manner.

7391 **D.3.3.3.1.6.3 Effect on Receipt**

7392 On receipt of this command, the server shall take and store a snapshot with cause 10 (Manually
 7393 Triggered from Client) set in addition to the requested cause (see Table D-52).

7394

7395 D.3.3.3.1.7 GetSnapshot Command7396 This command is used to request snapshot data from the cluster server. See section D.3.4.5 for
7397 further details.7398 **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)

7399 **Figure D-44—GetSnapshot Command Payload**7400 **D.3.3.3.1.7.2 Payload Details**7401 **Earliest Start Time (mandatory):** A UTC Timestamp indicating the earliest time of a snapshot
7402 to be returned by a corresponding *Publish Snapshot* command. Snapshots with a time stamp
7403 equal to or greater than the specified *Earliest Start Time* shall be returned.7404 **Latest End Time (mandatory):** A UTC Timestamp indicating the latest time of a snapshot to be
7405 returned by a corresponding *Publish Snapshot* command. Snapshots with a time stamp less than
7406 the specified *Latest End Time* shall be returned.7407 **Snapshot Offset (mandatory):** Where multiple snapshots satisfy the selection criteria specified
7408 by the other fields in this command, this field identifies the individual snapshot to be returned.
7409 An offset of zero (0x00) indicates that the first snapshot satisfying the selection criteria should be
7410 returned, 0x01 the second, and so on.7411 **Snapshot Cause (mandatory):** This field is used to select only snapshots that were taken due to
7412 a specific cause. The allowed values are listed in Table D-52. Setting this field to 0xFFFFFFFF
7413 indicates that all snapshots should be selected, irrespective of the cause.7414 **D.3.3.3.1.7.3 Effect on Receipt**7415 On receipt of this command, the server shall respond with one or more *Publish Snapshot*
7416 commands representing the first (or next) snapshot meeting the selection criteria and *Snapshot*
7417 *Offset* value detailed in this command. Details of the *Publish Snapshot* command are detailed in
7418 sub-clause D.3.2.3.1.7.7419 A ZCL Default Response with status NOT_FOUND shall be returned if the server does not have
7420 a snapshot which satisfies the received parameters (e.g. no snapshot with a timestamp between
7421 the *Earliest Start Time* and the *Latest End Time*).

7422

7423 D.3.3.3.1.8 StartSampling Command

7424 The sampling mechanism allows a set of samples of the specified type of data to be taken,
 7425 commencing at the stipulated start time. This mechanism may run concurrently with the
 7426 capturing of profile data, and may refer to the same parameters, albeit possibly at a different
 7427 sampling rate.

7428 **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)

7429 **Figure D-45– StartSampling Command Payload**

7430 **D.3.3.3.1.8.2 Payload Details**

7431 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. When
 7432 new information is provided that replaces older information for the same time period, this field
 7433 allows devices to determine which information is newer. The value contained in this field is a
 7434 unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)
 7435 identifying when the command was issued. Thus, newer information will have a value in the
 7436 Issuer Event ID field that is larger than older information.

7437 **StartSamplingTime (mandatory):** A UTC Time field to denote the time at which the sampling
 7438 should start. A start Date/Time of 0x00000000 shall indicate that the command should be
 7439 executed immediately. A start Date/Time of 0xFFFFFFFF shall cause an existing *StartSampling*
 7440 command with the same *Issuer Event ID* to be cancelled.

7441 **SampleType (mandatory):** An 8 bit enumeration that identifies the type of data being sampled.
 7442 Possible values are defined in Table D-54.

7443 **SampleRequestInterval (mandatory):** An unsigned 16-bit field representing the interval or
 7444 time in seconds between samples.

7445 **MaxNumberofSamples (mandatory):** A 16 bit unsigned integer that represents the number of
 7446 samples to be taken.

7447 **D.3.3.3.1.8.3 Effect on Receipt**

7448 On receipt of the *StartSampling* command, the Metering Device shall respond with a
 7449 *StartSamplingResponse* command indicating the *Sample ID* allocated to this Sampling session. If
 7450 the Metering Device is unable to support a further Sampling session, *Sample ID* shall be
 7451 returned as 0xFFFF. See D.3.2.3.1.14 for further details.

7452

7453 D.3.3.3.1.9 GetSampledData Command

7454 This command is used to request sampled data from the server. Note that it is the responsibility
 7455 of the client to ensure that it does not request more samples than can be held in a single command
 7456 payload.

7457 **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)

7458 **Figure D-46– GetSampledData Command Payload**7459 **D.3.3.3.1.9.2 Payload Details**

7460 **SampleID (mandatory):** Unique identifier allocated to this Sampling session. This field allows
 7461 devices to match response data with the appropriate request. See D.3.2.3.1.14 for further
 7462 details.**EarliestSampleTime (mandatory):** A UTC Timestamp indicating the earliest time of a
 7463 sample to be returned. Samples with a timestamp equal to or greater than the specified
 7464 *EarliestSampleTime* shall be returned.

7465 **SampleType (mandatory):** An 8 bit enumeration that identifies the required type of sampled
 7466 data. Possible values are defined in Table D-54.

7467 **NumberOfSamples (mandatory):** Represents the number of samples being requested, This
 7468 value cannot exceed the size stipulated in the *MaxNumberOfSamples* field in the *StartSampling*
 7469 command. If more samples are requested than can be delivered, the *GetSampledDataResponse*
 7470 command will return the number of samples equal to the *MaxNumberOfSamples* field. If fewer
 7471 samples are available for the time period, only those available are returned.

7472 **D.3.3.3.1.9.3 Effect on Receipt**

7473 On receipt of this command, the server shall respond with a *GetSampledDataResponse* command
 7474 containing the samples meeting the selection criteria detailed in this command. Details of the
 7475 *GetSampledDataResponse* command are detailed in sub-clause D.3.2.3.1.8.

7476 A ZCL Default Response with status NOT_FOUND shall be returned if the server does not have
 7477 sample data which satisfies the received parameters.

7478

7479 D.3.3.3.1.10 MirrorReportAttributeResponse Command

7480 This command is sent in response to the *ReportAttribute* command when the *MirrorReporting*
7481 attribute is set.

7482 **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)

7483 **Figure D-47 – MirrorReportAttributeResponse Command Payload**

7484 **D.3.3.3.1.10.2 Payload Details**

7485 The payload of this command is defined within the *ConfigureNotificationScheme* command.

7486 **Notification Scheme (mandatory):** see section D.3.3.2.1.3.

7487 **Notification Flags #N (mandatory):** see sections D.3.3.2.1.1 and D.3.3.2.1.2.

7488 **D.3.3.3.1.10.3 When Generated**

7489 The *MirrorReportAttributeResponse* command is generated in response to the *ReportAttribute*
7490 command when the *MirrorReporting* attribute is set. The *MirrorReportAttributeResponse*
7491 command is sent from the Mirror to the meter.

7492 **D.3.3.3.1.10.4 Effect on Receipt**

7493 On receipt of the *MirrorReportAttributeResponse*, the meter shall check the flags contained
7494 within the payload. It is then up to the meter to request any information that is waiting on the
7495 ESI.

7496

7497 D.3.3.3.1.11 ResetLoadLimitCounter Command

7498 The *ResetLoadLimitCounter* command shall cause the *LoadLimitCounter* attribute to be reset
7499 (see D.3.2.2.7.7 for further details).

7500 **D.3.3.3.1.11.1 Payload Format**

7501

Octets	4	4
Data Type	Unsigned 32-bit Integer	Unsigned 32-bit Integer
Field Name	Provider ID (M)	Issuer Event ID (M)

7502

Figure D-48– ResetLoadLimitCounter Command Payload

7503

D.3.3.3.1.11.2 Payload Details7504
7505

Provider ID (mandatory): An unsigned 32-bit field containing a unique identifier for the commodity provider.

7506
7507
7508
7509
7510

Issuer Event ID (mandatory): Unique identifier generated by the commodity provider. This field allows devices to determine if a new command has been issued. 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, a newer command will have a value in the Issuer Event ID field that is larger than previous versions of the command.

7511

7512

D.3.3.3.1.12 Change Supply Command7513
7514

This command is sent from the Head-end or ESI to the Metering Device to instruct it to change the status of the valve or load switch, i.e. the supply.

7515

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
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

7516

Figure D-49– Change Supply Command Payload

7517

D.3.3.3.1.12.2 Payload Details7518
7519

Provider ID (mandatory): An unsigned 32-bit field containing a unique identifier for the commodity provider to whom this command relates.

7520
7521
7522
7523
7524
7525

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.

7526
7527

Request Date/Time (mandatory): A UTC Time field to indicate the date and time at which the supply change was requested.

Implementation Date/Time (mandatory): A UTC Time field to indicate the date at which the supply change is to be applied. An *Implementation Date/Time* of 0x00000000 shall indicate that the command should be executed immediately. An *Implementation Date/Time* of 0xFFFFFFFF shall cause an existing but pending *Change Supply* command with the same *Provider ID* and *Issuer Event ID* to be cancelled (the status of the supply will not change but the *Proposed Change Supply Implementation Time* attribute shall be reset to zero).

Proposed Supply Status (after Implementation): An 8-bit enumeration field indicating the status of the energy supply controlled by the Metering Device following implementation of this command. The enumerated values for this field are outlined in Table D-56.

Supply Control Bits: An 8-bit BitMap where the least significant nibble defines the Supply Control bits, the encoding of which is outlined in Table D-66:

Table D-66— Supply Control Bits

Bits	Description
0	Acknowledge Required
1	Reserved
2	Reserved
3	Reserved

7540

Acknowledge Required: Indicates that a *Supply Status Response* command is to be sent in response to this command. Note that the *Supply Status Response* command will only be returned to the originator when the *Change Supply* command has been successfully executed.

7544 D.3.3.3.1.12.3 When Generated

7545 A Head-end or ESI may send an INTERRUPT, ARM or (if allowed) RESTORE command to a
7546 metering device.

The execution of an INTERRUPT or ARM command may be delayed, as indicated by the *Implementation Date/Time* field; these commands shall only come from a Head-End via an ESI. A subsequent command with a new *Implementation Date/Time* shall override an existing delayed command. A new command with an *Implementation Date/Time* of 0x00000000 shall be executed immediately, but shall not cancel an existing delayed command; to override an existing delayed command with a command to be executed immediately, a command to cancel the existing command should first be sent followed by the new command to be executed immediately (see notes on *Implementation Date/ Time* field in D.3.3.3.1.12.2 for further details).

7555 The addition of credit or selection of Emergency credit shall not cause a delayed INTERRUPT
7556 command to be cancelled (these will be cancelled by the Head-End and a new supply control
7557 command sent down).

7558 **D.3.3.3.1.12.4 Effect on Receipt**

7559 If required, a *Supply Status Response* command shall be returned to the originator when the
 7560 *Change Supply* command has been successfully executed (see D.3.2.3.1.13 for further details).

7561 A ZCL response, indicating ‘Unauthorized’ (NOT_AUTHORIZED), shall be immediately
 7562 returned to an originator requesting a supply change that is not allowed in the current application.

7563 A ZCL response, indicating ‘Unavailable’ (UNSUP_CLUSTER_COMMAND), shall be
 7564 immediately returned to an originator requesting a supply change by a metering device that is
 7565 incapable of carrying out the action (e.g. an INTERRUPT command to a metering device that has
 7566 no contactor).

7567 A ZCL response, indicating INVALID_VALUE, shall be immediately returned to an originator
 7568 requesting a supply change containing a non-zero Implementation Date/Time that is less than or
 7569 equal to the current date/time (i.e. is in the past).

7570

7571 **D.3.3.3.1.13 Local Change Supply Command**

7572 This command is a simplified version of the *Change Supply* command, intended to be sent from
 7573 an IHD to a meter as the consequence of a user action on the IHD. Its purpose is to provide a
 7574 local disconnection/reconnection button on the IHD in addition to the one on the meter.

7575 **D.3.3.3.1.13.1 Payload Format**

Octets	1
Data Type	8 bits Enumeration
Field Name	Proposed Supply Status

7576 **Figure D-50– Local Change Supply Command Payload**

7577 **D.3.3.3.1.13.2 Payload Details**

7578 **Proposed Supply Status:** An 8-bit enumeration field indicating the status of the energy supply
 7579 controlled by the Metering Device following implementation of this command. The enumerated
 7580 values for this field are outlined in Table D-67:

7581 **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

7582 **D.3.3.3.1.13.3 When Generated**

7583 An IHD may only request an OFF/ARMED or ON status for the supply. This corresponds to a
 7584 local disconnection or reconnection (from Armed state) of the supply, similar to what can be
 7585 achieved with a button normally present on electricity meters equipped with a contactor.

7586 **D.3.3.3.1.13.4 Effect on Receipt**

7587 No *Supply Status Response* command shall be returned to the originator.

7588 A ZCL response, indicating ‘Unauthorized’ (NOT_AUTHORIZED), shall be immediately
 7589 returned to an originator requesting a supply change that is not allowed in the current application.

7590 A ZCL response, indicating ‘Unavailable’ (UNSUP_CLUSTER_COMMAND), shall be
 7591 immediately returned to an originator requesting a supply change by a metering device that is
 7592 incapable of carrying out the action (e.g. an INTERRUPT command to an electricity meter that
 7593 has no contactor or to a gas meter for which this command is not allowed).

7594

7595 D.3.3.3.1.14 SetSupplyStatus Command

7596 This command is used to specify the required status of the supply following the occurrence of
 7597 certain events on the meter. The meter shall check these requirements to understand whether the
 7598 supply should be disabled or enabled following one of these events.

7599 **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):

7600 **Figure D-51– SetSupplyStatus Command Payload**

7601 **D.3.3.3.1.14.2 Payload Details**

7602 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. When
 7603 new information is provided that replaces older information for the same time period, this field
 7604 allows devices to determine which information is newer. The value contained in this field is a
 7605 unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)
 7606 identifying when the command was issued. Thus, newer information will have a value in the
 7607 *Issuer Event ID* field that is larger than older information.

7608 **SupplyTamperState (mandatory):** The *SupplyTamperState* indicates the required status of the
 7609 supply following the detection of a tamper event within the metering device. The enumerated
 7610 values for this field are outlined in Table D-68.

SupplyDepletionState (mandatory): The *SupplyDepletionState* indicates the required status of the supply following detection of a depleted battery within the metering device. The enumerated values for this field are outlined in Table D-68.

SupplyUncontrolledFlowState (mandatory): The *SupplyUncontrolledFlowState* indicates the required status of the supply following detection of an uncontrolled flow event within the metering device. The enumerated values for this field are outlined in Table D-68.

LoadLimitSupplyState (mandatory): The *LoadLimitSupplyState* indicates the required status of the supply once the device is in a load limit state. The enumerated values for this field are outlined in Table D-68.

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

7621
7622 D.3.3.3.1.15 [SetUncontrolledFlowThreshold Command](#)
7623 This command is used to update the ‘Uncontrolled Flow Rate’ configuration data used by flow
7624 meters.

7625 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)

Octets	1	2
Data Type	Unsigned 8 bit Integer	Unsigned 16 bit Integer

Octets	1	2
Field Name	Stabilisation Period (M)	Measurement Period (M)

Figure D-52– SetUncontrolledFlowThreshold Command Payload

D.3.3.3.1.15.2 Payload Details

Provider ID (mandatory): An unsigned 32-bit field containing a unique identifier for the commodity provider to whom this command relates.

Issuer Event ID (mandatory): Unique identifier generated by the commodity provider. When new information is provided that replaces older information for the same time period, this field allows devices to determine which information is newer. The value contained in this field is a unique number managed by upstream servers or a UTC based time stamp (UTCTime data type) identifying when the command was issued. Thus, newer information will have a value in the *Issuer Event ID* field that is larger than older information.

Uncontrolled Flow Threshold (mandatory): The threshold above which a flow meter (e.g. Gas or Water) shall detect an uncontrolled flow. A value of 0x0000 indicates the feature in unused.

Unit of Measure (mandatory): An enumeration indicating the unit of measure to be used in conjunction with the *Uncontrolled Flow Threshold* attribute. The enumeration used for this field shall match one of the UnitOfMeasure values using a pure binary format as defined in the Metering cluster (see sub-clause D.3.2.2.4.1).

Multiplier (mandatory): An unsigned 16-bit value indicating the multiplier, to be used in conjunction with the *Uncontrolled Flow Threshold* and *Divisor* fields, to determine the true flow threshold value. A value of 0x0000 is not allowed.

Divisor (mandatory): An unsigned 16-bit value indicating the divisor, to be used in conjunction with the *Uncontrolled Flow Threshold* and *Multiplier* fields, to determine the true flow threshold value. A value of 0x0000 is not allowed.

7649 Stabilisation Period (mandatory): An unsigned 8-bit value indicating the time given to allow
7650 the flow to stabilize. It is defined in units of tenths of a second.

Measurement Period (mandatory): An unsigned 16-bit value indicating the period over which the flow is measured and compared against the *Uncontrolled Flow Threshold* value. It is defined in units of 1 second.

7654
7655
7656

7657 **D.3.4 Metering Application Guidelines**7658 **D.3.4.1 Attribute Reporting**

7659 Attribute reporting may be used for sending information in the Reading Information,
7660 TOU Information, Meter Status, and Historical Consumption attribute sets. Use of the
7661 *Report Attribute* command without report configuration may be used for unsolicited notification
7662 of an attribute value change. Sleepy devices may have to poll.

7663 **D.3.4.2 Fast Polling or Reporting for Monitoring Energy Savings**

7664 Client devices, such as an energy gateway, smart thermostat, or in-home displays can monitor
7665 changes to energy saving settings within the premises and give users near real time feedback and
7666 results. The Metering cluster can support this by using Attribute Reporting and sending updates
7667 at a much faster rate for a short period of time. Client devices can also perform a series of
7668 Attribute reads to accomplish the same task. In either case, requests or updates shall be limited
7669 to a maximum rate of once every two seconds for a maximum period of 15 minutes. These
7670 limitations are required to ensure Smart Energy profile based devices do not waste available
7671 bandwidth or prevent other operations within the premises.

7672 **D.3.4.3 Metering Data Updates**

7673 The frequency and timeliness of updating metering data contained in the Metering
7674 Cluster attributes and Profile Intervals is up to the individual Metering device manufacturer's
7675 capabilities. As a best practice recommendation, updates of the metering data should not cause
7676 delivery of the information to end devices more often than once every 30 seconds. End
7677 devices should also not request information more often than once every 30 seconds. The Fast
7678 Polling attributes and commands shall be used by client devices requesting information more
7679 often than once every 30 seconds.

7680 **D.3.4.3.1 Fast Polling Periods**

7681 Since the *DefaultUpdatePeriod* specifies the normal update interval and *FastPollUpdatePeriod*
7682 specifies the fastest possible update interval, it is recommended that metering clients read these
7683 attributes to determine the optimal normal/fast polling interval and the optimal fast poll period
7684 to request. Client devices shall not request data more frequent than *FastPollUpdatePeriod* or
7685 the *AppliedUpdatePeriod*.

7686 **D.3.4.4 Mirroring**

7687 The SE Profile specifies Mirror support in the Metering cluster to store and provide access to
7688 data from metering devices on battery power. Devices with resources to support mirroring
7689 advertise the capability using the Basic Attribute Physical Environment.

7690 D.3.4.4.1 **Discovery**

7691 The SE standard does not prescribe how Mirroring is implemented. Devices may query the
7692 Basic Cluster attribute *PhysicalEnvironment* to determine Mirrored device capacity prior to
7693 CBKE (see sub-clause D.3.4.4.2 below). This would allow a battery based end device to
7694 discover if an ESI has capacity to mirror data prior to the process of joining the network in a
7695 secure manner, thereby reducing retry attempts. This would also enhance the service discovery
7696 of the ZDO Match Descriptor that would be used to determine if an endpoint can request the
7697 setup and removal of a mirrored Metering cluster. Once a device has joined the network and
7698 performed CBKE, it can then request setup of a mirrored metering cluster. ZDO Discovery
7699 should be supported to allow HAN devices to discover the mirror endpoints; only active mirror
7700 endpoints shall be discoverable. This process may need to be repeated in the case of a Trust
7701 Center swap-out (refer to sub-clause 5.4.2.2.3 for further information).

7702 D.3.4.4.2 **Mirror Attributes**

7703 The mandatory *Basic*, *Metering*, and (where applicable) *Prepayment* attributes shall be
7704 supported. The Basic Cluster *PhysicalEnvironment* attribute shall be supported on ESIs
7705 supporting mirroring functionality; an enumerated value of 0x01 would indicate that the
7706 device has the capacity to mirror an end device; a value of 0x00 would specify an “Unspecified
7707 environment” per the ZCL specification. Only the Basic cluster for devices capable of
7708 providing a mirror shall have the *PhysicalEnvironment* attribute set to 0x01. The *ZCL Report*
7709 *Attribute* command shall be used to push data to the mirror. Only the metering device that has
7710 been granted a mirror on a certain endpoint is allowed to push data to that endpoint. The ZCL Not
7711 Authorized return status shall be used to provide access control. The use of ZCL Report
7712 Configuration shall not be required to generate *Report Attribute Command*.

7713 Manufacturers will design and manufacture devices to meet customer requirement specifications
7714 that will state the functionality of the battery powered meter and therefore devices supporting
7715 mirroring in the field will also have to support those requirements through an appropriate choice
7716 of optional attributes. Battery powered devices will report attributes to the mirror as required by
7717 the customer specification. In the event that the mirror is out of memory space or cannot
7718 support the attribute it shall respond ATTRIBUTE_UNSUPPRESSED back to the battery-
7719 powered meter. The same response (ATTRIBUTE_UNSUPPRESSED) will be sent to a device
7720 querying the mirror for an attribute it doesn't support. A device querying the mirror for an
7721 attribute that is supported but not yet available (the battery powered meter hasn't yet sent the
7722 attribute) shall receive a response ATTRIBUTE_UNAVAILABLE from the mirror.

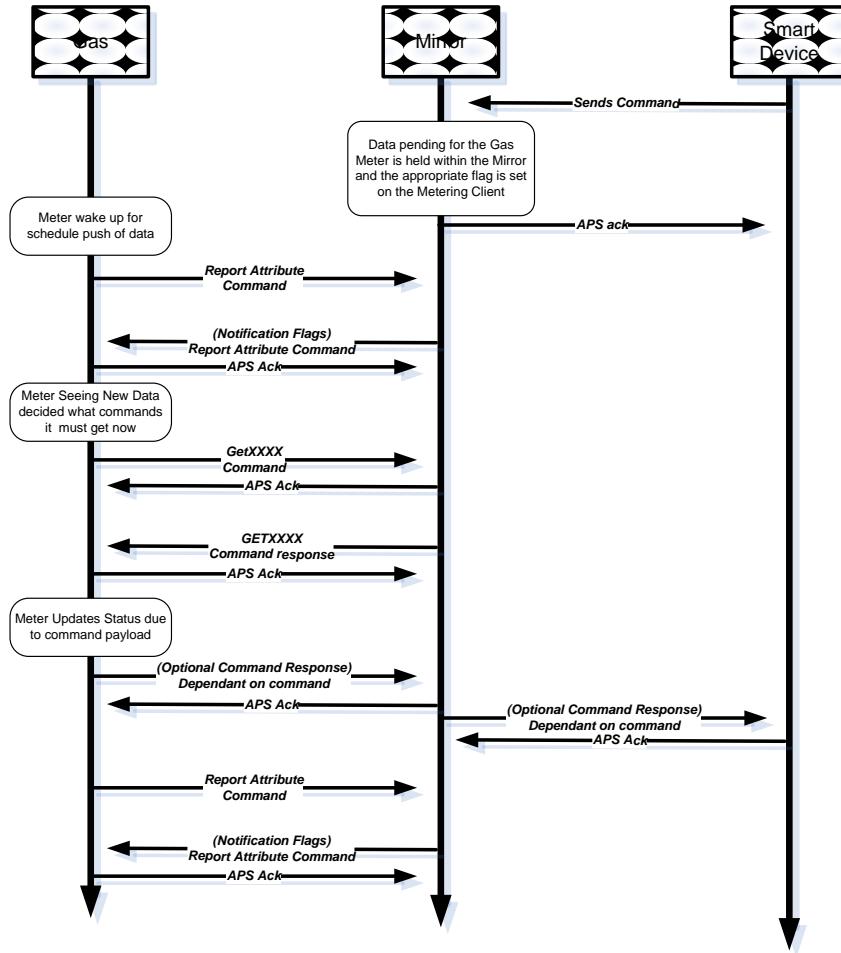
7723 D.3.4.4.3 **Two Way Mirror for BOMD**

7724 The primary purpose of a mirror is to present data from a sleepy battery operated mirrored
7725 device (BOMD), to a HAN, when communication to the BOMD is not available. However, there
7726 is also a need to pass data to the BOMD in these circumstances.

7727 Any device on the HAN wishing to communicate with a BOMD must do so via the mirror. The
7728 mirror and the BOMD must support the Notification Attribute Set of the Metering cluster,
7729 designed to allow the BOMD to establish if there are any messages waiting on the mirror for
7730 collection. There are 4 mechanisms provided to allow information destined for a BOMD to be
7731 transferred to the BOMD:-

- 7732 1 For several required actions, the Notification Flag conveys all required information. Many of
7733 the bits within the *Functional Notification Flags* attribute utilize this method; the *Push All*
7734 *Static Data - Metering Cluster* bit is an example of this.
 - 7735 2 For those clusters where the BOMD is a client (e.g. Price, Calendar, Device Management),
7736 the flags in the Notification Attribute Set allow the BOMD to quickly determine if there is
7737 any new information of interest to the BOMD. This information is normally sent from the
7738 backhaul (i.e. the Head End System). Upon waking, and having acquired the status of the
7739 relevant Notification Flags, the BOMD will fetch those commands matching the set flags by
7740 sending the appropriate ‘get’ command(s) **to the associated ESI endpoint**. (Note that the
7741 associated data will usually be held on the device providing the ESI; therefore the associated
7742 get/publish commands could be utilized multiple times). It is recommended that a ‘binding-
7743 type’ mechanism is used internally within the mirroring device to link the ESI and Mirror
7744 endpoints.
 - 7745 3 Where a cluster server is located on the BOMD (e.g. Metering, Prepayment and Basic
7746 clusters), pre-specified transient commands sent from cluster clients will have to be buffered
7747 on the Mirror until such time as the BOMD awakes and can fetch them. Upon waking, and
7748 having acquired the status of the relevant Notification Flags, the BOMD will fetch those
7749 commands matching the set flags by sending a *GetNotifiedMessage* command **to the Mirror**
7750 **endpoint**. In this case, the Mirror shall remember the address of the device initially
7751 originating the command, so that any response can be returned, via the mirror, to that device.
7752 As an example, the handling of a (Prepayment cluster) *Consumer Top Up* command utilizes
7753 this method.
 - 7754 4 Non-specified transient commands, destined for cluster servers located on a BOMD, will
7755 also have to be buffered on the Mirror until such time as the BOMD awakes. In this case, the
7756 appropriate ‘Stay Awake’ Notification flag will be set (to advise the BOMD to remain
7757 awake for a longer period) and, once the mirroring device recognizes that the BOMD is
7758 awake, it shall attempt to push those buffered commands to the BOMD as soon as possible.
7759 Attempts to transfer these commands shall be repeated until such time as the command(s)
7760 is/are successfully moved to the BOMD (this may not be within the same BOMD wake
7761 period). The Mirror shall again remember the address of the device initially originating the
7762 command, so that any response can be returned, via the mirror, to that device. As an
7763 example, the handling of a (Metering cluster) *Get Profile* command utilizes this method.
- 7764 .

7765



7766

7767

Figure D-53– Example of Data flow from IHD to Gas meter

7768 The example in Figure D-53 shows how data is transferred from a HAN device (e.g. IHD) to a
 7769 BOMD (e.g. gas meter) via the mirror. There are a number of commands that will be sent from
 7770 the IHD to meter, for example:

- 7771 • Credit Top Up
- 7772 • Emergency Credit Select
- 7773 • Local Change Supply

7774 The sequence of events is as follows:

- 7775 1 The IHD sends a command to the Mirror

- 7776 **2** The mirror “caches” the command and sets the appropriate notification flag, to signal that
7777 data is waiting. The mirror also returns a ZCL Default Response to the initiating device with
7778 a status code of NOTIFICATION_PENDING. If the command buffer on the mirror is
7779 already full, the mirror shall instead return a ZCL Default Response to the initiating device
7780 with a status code of INSUFFICIENT_SPACE.
- 7781 **3** The gas meter wakes up and polls for a notification
- 7782 **4** The notification is returned
- 7783 **5** This may be all that is required (e.g. a request to update static data on the mirror).
- 7784 **6** The meter requests the data according to the Notification flag that was raised
- 7785 **7** The mirror sends the command that was originally received from the IHD
- 7786 **8** The meter may update data on the mirror in order to indicate to the device initiating the
7787 command that its action has been carried out.

7788

7789 D.3.4.4.3.1 Responses to an Initiating Device

7790 Commands that have been buffered on a Mirror may trigger ZCL Default or command-specific
7791 responses to be returned once the command has reached and been actioned by the BOMD. In
7792 turn, these responses should be communicated to the device originally initiating the buffered
7793 command.

7794 In order to ensure that these responses are correctly relayed back to the device initiating the
7795 original command, it is recommended that:-

- 7796 **1** The mirroring device, if supporting Two-way Mirroring, be able to store information that
7797 can be used to track the originator of a command stored in the buffer when the command is
7798 retrieved by the BOMD. This information should include, as a minimum, the device address,
7799 endpoint and Transaction Sequence Number (TSN) of the original command, for the
7800 purposes of relaying a response back to the originator. The TSN used when the buffered
7801 command is forwarded to the BOMD should also be included in the stored information for
7802 the purpose of matching the information to any associated response. A TSN in the ZCL
7803 header is only 8 bits and this may not provide enough information to produce a unique ID (or
7804 unique enough); if this is considered to be insufficient information, the mirroring device
7805 could also store the cluster and command id.
- 7806 **2** The mirroring device should relay the ZCL payload of the buffered command to the BOMD
7807 using a newly generated ZCL command (i.e. using the address and security associated with
7808 the BOMD).
- 7809 **3** The payload of a response from the BOMD to the retrieved buffered command, should be
7810 relayed to the originating device in a newly generated ZCL response which utilizes the

7811 information stored by the mirroring device as detailed in point 1. This is to include the TSN
 7812 of the command received from the originating HAN device.

- 7813 4 All commands retrieved by the BOMD using the notification flag mechanism should support
 7814 a ZCL default response in cases where an explicit response is not defined (this is the normal
 7815 mode of operation; however the default response can be disabled).
 7816

7817 **D.3.4.4.3.2 Predefined Notification Scheme A**

7818 Notification Scheme A is a predefined scheme for the order of the bit strings within each of the
 7819 *NotificationFlag#N* attributes. See sections D.3.2.3.1.10 and D.3.2.3.1.11 for configuration of
 7820 other schemes. Refer to section D.3.4.4.3 for details on the usage of these Notification Flags.

7821 ***D.3.4.4.3.2.1 MirrorReportAttributeResponse Command Format***

7822 The format for Notification Scheme A is ***0x0FFFFFFF*** meaning that the first and only
 7823 Notification flag to be transmitted within the *MirrorReportAttributeResponse* command will be
 7824 the *FunctionalNotificationFlags* attribute.

7825 ***FunctionalNotificationFlags Attribute:*** Defined in section D.3.3.2.1.1.
 7826

7827 **D.3.4.4.3.3 Predefined Notification Scheme B**

7828 Notification Scheme B is a predefined scheme for the order of the bit strings within each of the
 7829 *NotificationFlag#N* attributes. See sections D.3.2.3.1.10 and D.3.2.3.1.11 for configuration of
 7830 other schemes. Refer to section D.3.4.4.3 for details on the usage of these Notification Flags.

7831 ***D.3.4.4.3.3.1 MirrorReportAttributeResponse Command Format***

7832 The format for Notification Scheme B is ***0x01234FFF*** meaning the first Notification flag to be
 7833 transmitted within the *MirrorReportAttributeResponse* command will be the
 7834 *FunctionalNotificationFlags* attribute followed by *NotificationFlags2* to *NotificationFlags5*.

7835 ***FunctionalNotificationFlags Attribute:*** Defined in section D.3.3.2.1.1.

7836 ***NotificationFlags2 Attribute:*** The *NotificationFlags2* attribute shall be configured to support
 7837 the Price cluster and is implemented as a set of bit flags which are defined as shown below:
 7838

7839 **Table D-69– Notification Flags 2**

Bit Number	Waiting Command
0	PublishPrice ^a
1	PublishBlockPeriod

Bit Number	Waiting Command
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

7840 ^a A *Publish Price* command may result from more than one ‘Get’ command; for clarity, a *GetCurrentPrice* command
 7841 should be sent when this flag is set, and a *GetScheduledPrices* command MAY also be sent.
 7842

7843 **NotificationFlags3 Attribute:** The *NotificationFlags3* attribute shall be configured to support
 7844 the Calendar cluster and is implemented as a set of bit flags which are defined as shown below:

7845 **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

7846

7847 **NotificationFlags4 Attribute:** The *NotificationFlags4* attribute shall be configured to support
 7848 the Prepayment cluster and is implemented as a set of bit flags which are defined as shown
 7849 below:

7850 **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

7851

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:

7855

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

7856

7857

7858 **NotificationFlags6 Attribute:** This attribute is not supported, with no bits set.

7859 **NotificationFlags7** Attribute: This attribute is not supported, with no bits set.

7860 **NotificationFlags8** Attribute: This attribute is not supported, with no bits set.

7861

7862 **D.3.4.5 An Introduction to Snapshots**

7863 Where a permanent back-haul connection is not guaranteed, there are occasions when the values
7864 of data items need to be frozen for purposes such as consumer billing. The Snapshot mechanism
7865 is provided to satisfy this requirement.

7866 Snapshots can be triggered in a number of ways:-

- 7867 • Automatically as a result of certain activities (e.g. end of billing period, change of tariff,
7868 change of supplier)
- 7869 • At pre-defined points using the *ScheduleSnapshot* command (and confirmed via a
7870 *ScheduleSnapshotResponse* command)
- 7871 • As a manual/one-off action using the *TakeSnapshot* command (and confirmed via a
7872 *TakeSnapshotResponse* command)

7873 Details of stored Snapshots can be requested using the *GetSnapshot* command; the content(s) of
7874 the required Snapshot(s) will then be returned using one or more *Publish Snapshot* commands.

7875

7876 **D.3.4.6 Supply Control**

7877 The Supply Control functionality allows a Head-end System to remotely control the status of the
7878 valve or contactor within a meter. The states of supply status are necessary due to the safety
7879 requirements in certain countries, these are:

- 7880 • ON
- 7881 • OFF
- 7882 • ARMED

7883 The ARMED state is to allow for a remote restoration of the supply that requires action by the
7884 consumer (such as pressing a button on the meter or the IHD). This is to ensure the supply is not
7885 restored remotely whilst in an unsafe situation. The three corresponding commands derived from
7886 IEC 62055 are:

- 7887 • RESTORE
- 7888 • INTERRUPT
- 7889 • ARM

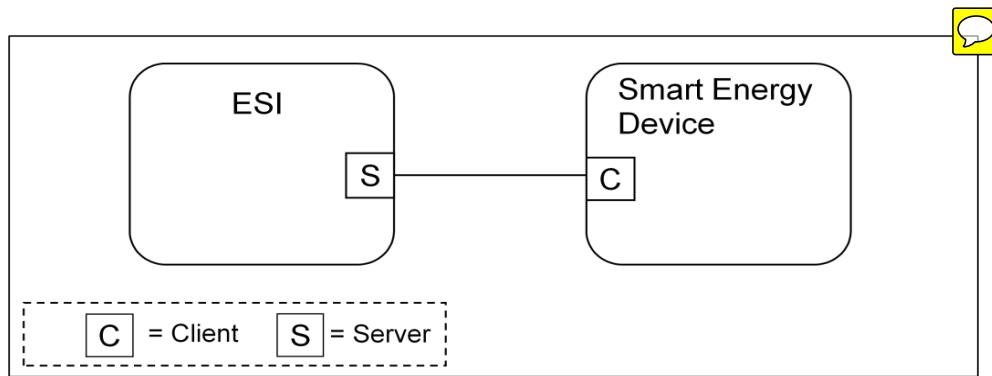
7891

7892

7893 **D.4 Price Cluster**7894 **D.4.1 Overview**

7895 The Price Cluster provides the mechanism for communicating Gas, Energy, or Water pricing
7896 information within the premises. This pricing information is distributed to the ESI from either the
7897 utilities or from regional energy providers. The ESI conveys the information (via the Price Cluster
7898 mechanisms) to other Smart Energy devices.

7899



7900

Figure D-54– Price Cluster Client Server Example

7902 Please note the ESI is defined as the Server due to its role in acting as the proxy for upstream price
7903 management systems and subsequent data stores.

7904 **D.4.2 Server**7905 **D.4.2.1 Dependencies**

7906 Events carried using this cluster include a timestamp with the assumption that target devices
7907 maintain a real time clock. Devices can acquire and synchronize their internal clocks via the ZCL
7908 Time server.

7909 If a device does not support a real time clock it is assumed that the device will interpret and
7910 utilize the “Start Now” value within the Time field.

7911 **D.4.2.2 Attributes**

7912 For convenience, the attributes defined in this cluster are arranged into sets of related
7913 attributes; each set can contain up to 256 attributes. Attribute identifiers are encoded such that
7914 the most significant Octet specifies the attribute set and the least significant Octet specifies the
7915 attribute within the set. The currently defined attribute sets are listed in the following Table

7916 D-73. The Price Cluster is broken down in to Delivered attribute sets 0x00 to 0x7F and Received
 7917 attribute sets 0x80 to 0xFF.

7918 *Note: Price Cluster Attribute Sets 0x01-0x04 in this revision of this specification are provisional*
 7919 *and not certifiable. This feature set may change before reaching certifiable status in a future*
 7920 *revision of this specification.*

7921

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

7922

7923 D.4.2.2.1 Tier Label (Delivered) Set

7924 *Note: Tier Labels 0x0006-0x000E in this revision of this specification are provisional*
 7925 *and not certifiable. This feature set may change before reaching certifiable status in a future*
 7926 *revision of this specification.*

7927

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					

7930 D.4.2.2.1.1 TierNPriceLabel Attributes

7931 The *TierNPriceLabel* attributes provide a method for utilities to assign a label to the Price Tier
 7932 declared within the *Publish Price* command. The *TierNPriceLabel* attributes are a ZCL Octet
 7933 String field capable of storing a 12 character string (the first Octet indicates length) encoded in
 7934 the UTF-8 format. Example Tier Price Labels are “Normal”, “Shoulder”, “Peak”, “Real Time”
 7935 and “Critical”. There are 48 Tier Labels.

7936 Although not prohibited, it is likely (and allowed) that a server will reject an attempt to write to
 7937 these attributes; if rejected, the server shall return a ZCL Default Response with a status of either
 7938 NOT_AUTHORIZED or READ_ONLY. A client should make provision for a write attempt to
 7939 be rejected.⁵
 7940

7941 D.4.2.2.2 **Block Threshold (Delivered) Set**

7942 The following set of attributes provides remote access to the Price server Block Thresholds. 
 7943 Block Threshold values are crossed when the CurrentBlockPeriodConsumptionDelivered
 7944 attribute value is greater than a *BlockNThreshold* attribute. The number of block thresholds is
 7945 indicated by the *Number of Block Thresholds* field in the associated *Publish Price* command.
 7946 The number of blocks is one greater than the number of thresholds.

7947 *Note: The Block Cluster Attribute Set in this revision of this specification is provisional and
 7948 not certifiable. This feature set may change before reaching certifiable status in a future
 7949 revision of this specification.*

7950

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

⁵ CCB 1511

0x0104	<i>Block5Threshold</i>	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
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

7951

7952 D.4.2.2.2.1 BlockNThreshold7953 Attributes *Block1Threshold* through *Block15Threshold* represent the block threshold values for a
7954 given period (typically the billing cycle). These values may be updated by the utility on a

7955 seasonal or annual basis. The thresholds are established such that crossing the threshold of
 7956 energy consumption for the present block activates the next higher block, which can affect the
 7957 energy rate in a positive or negative manner. The values are absolute and always increasing.
 7958 The values represent the threshold at the end of a block. The Unit of Measure will be based on the
 7959 fields defined in the *Publish Price* command, the formatting being defined by attributes within
 7960 the *Block Period* attribute set.

7961 D.4.2.2.2 BlockThresholdCount Attribute

7962 Where a single set of thresholds is used, the *BlockThresholdCount* attribute indicates the number
 7963 of applicable *BlockNThresholds*. Where more than one set of thresholds is used, each set will be
 7964 accompanied by an appropriate *TierNBlockThresholdCount* attribute (see D.4.2.2.2.4).

7965 D.4.2.2.3 TierNBlockMThreshold Attributes

7966 Attributes *Tier1Block1Threshold* through *Tier15Block15Threshold* represent the block threshold
 7967 values applicable to a specific TOU tier for a given period (typically the billing cycle). These
 7968 values may be updated by the utility on a seasonal or annual basis. The thresholds are established
 7969 such that crossing the threshold of energy consumption for the present block activates the next
 7970 higher block, which can affect the energy rate in a positive or negative manner. The values are
 7971 absolute and always increasing. The values represent threshold at the end of a block. The Unit of
 7972 Measure will be based on the fields defined in the *Publish Price* command, the formatting being
 7973 defined by attributes within the *Block Period* attribute set.

7974 D.4.2.2.4 TierNBlockThresholdCount Attributes

7975 The *TierNBlockThresholdCount* attributes hold the number of block thresholds applicable to a
 7976 given tier. These attributes are used in the case when a combination (TOU/Hybrid) tariff has a
 7977 separate set of thresholds for each TOU tier. Unused *TierNBlockThresholdCount* attributes shall
 7978 be set to zero.

7979 D.4.2.2.3 **Block Period (Delivered) Set**

7980 The following set of attributes provides remote access to the Price server Block Threshold
 7981 period (typically the billing cycle) information.

7982 **Note:** The *Block Period Attribute Set* in this revision of this specification is provisional and
 7983 not certifiable. This feature set may change before reaching certifiable status in a future
 7984 revision of this specification.

7985

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 (minutes)</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 to 0x02FF	Reserved					

7986

7987 D.4.2.2.3.1 StartofBlockPeriod Attribute

7988 The *StartofBlockPeriod* attribute represents the start time of the current block tariff period. A
7989 change indicates that a new Block Period is in effect, see sub-clause D.4.4.3 for further details.

7990 D.4.2.2.3.2 BlockPeriodDuration Attribute

7991 The *BlockPeriodDuration* attribute represents the current block tariff period duration in minutes.
7992 A change indicates that only the duration of the current Block Period has been modified. A client
7993 device shall expect a new Block Period following the expiration of the new duration.

7994 D.4.2.2.3.3 ThresholdMultiplier Attribute

7995 *ThresholdMultiplier* provides a value to be multiplied against Threshold attributes. If present,
7996 this attribute must be applied to all Block Threshold values to derive values that can be
7997 compared against the *CurrentBlockPeriodConsumptionDelivered* attribute within the Metering
7998 cluster (see D.3.2.2.1.13). This attribute must be used in conjunction with the *ThresholdDivisor*
7999 attribute. An attribute value of zero shall result in a unitary multiplier (0x000001).

8000 D.4.2.2.3.4 ThresholdDivisor Attribute

8001 *ThresholdDivisor* provides a value to divide the result of applying the *ThresholdMultiplier*
8002 attribute to Block Threshold values to derive values that can be compared against the
8003 *CurrentBlockPeriodConsumptionDelivered* attribute within the Metering cluster (see
8004 D.3.2.2.1.13). This attribute must be used in conjunction with the *ThresholdMultiplier*
8005 attribute. An attribute value of zero shall result in a unitary divisor (0x000001).

8006 D.4.2.2.4 Commodity Set

8007 The following set of attributes represents items that are associated with a particular
8008 commodity.

8009 **Note:** The Commodity Attribute Set in this revision of this specification is provisional and
8010 not certifiable. This feature set may change before reaching certifiable status in a future
8011 revision of this specification.

8012

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					

8013

8014

8015 D.4.2.2.4.1 CommodityType Attribute 

8016 *CommodityType* provides a label for identifying the type of pricing server present. The attribute is
 8017 an enumerated value representing the commodity. The defined values are represented by the
 8018 non-mirrored values (0-127) in the *MeteringDeviceType* attribute enumerations (refer to Table
 8019 D-27).

8020 D.4.2.2.4.2 Standing Charge Attribute

8021 The value of the *Standing Charge* is a daily fixed charge associated with supplying the
 8022 commodity, measured in base unit of Currency with the decimal point located as indicated by
 8023 the Trailing Digits field of a *Publish Price* command (see sub-clause D.4.2.4.1) or
 8024 *PriceTrailingDigit* attribute (see D.4.2.2.7.7). A value of 0xFFFFFFFF indicates attribute not
 8025 used.

8026 D.4.2.2.4.3 ConversionFactor Attribute 

8027 The conversion factor is used for gas meter and takes into account changes in the volume of gas
 8028 based on temperature and pressure. The *ConversionFactor* attribute represents the current active
 8029 value. The *ConversionFactor* is dimensionless. The default value for the *ConversionFactor* is 1,
 8030 which means no conversion is applied. A price server can advertise a new/different value at any
 8031 time.

8032 D.4.2.2.4.4 ConversionFactorTrailingDigit Attribute

8033 An 8-bit BitMap used to determine where the decimal point is located in the *ConversionFactor*
 8034 attribute. The most significant nibble indicates the number of digits to the right of the decimal
 8035 point. The least significant nibble is reserved. The *ConversionFactorTrailingDigit* attribute
 8036 represents the current active value.

8037 D.4.2.2.4.5 CalorificValue Attribute 

8038 The amount of heat generated when a given mass of fuel is completely burned. The
 8039 *CalorificValue* is used to convert the measured volume or mass of gas into kWh. The
 8040 *CalorificValue* attribute represents the current active value.

8041 D.4.2.2.4.6 CalorificValueUnit Attribute

8042 This attribute defines the unit for the *CalorificValue*. This attribute is an 8-bit enumerated
 8043 field. The values and descriptions for this attribute are listed in Table D-78 below. The
 8044 *CalorificValueUnit* attribute represents the current active value.

8045 **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

8046

8047 D.4.2.2.4.7 CalorificValueTrailingDigit Attribute

8048 An 8-bit BitMap used to determine where the decimal point is located in the *CalorificValue*
 8049 attribute. The most significant nibble indicates the number of digits to the right of the decimal
 8050 point. The least significant nibble is reserved. The *CalorificValueTrailingDigit* attribute represents
 8051 the current active value.

8052 D.4.2.2.5 Block Price Information (Delivered) Set

8053 The following set of attributes provide remote access to the block prices. The Block Price
 8054 Information attribute set supports Block and combined Tier-Block pricing, the number of
 8055 blocks is one greater than the number of block thresholds defined in the Pricing cluster.

8056 **Note:** The Number of Block Thresholds field of the Publish Price command in this revision of this
 8057 specification are provisional and not certifiable. The specific use of this field may change
 8058 before reaching certifiable status in a future revision of this specification.

8059

Table D-79– Block Price Information Attribute Set

Identifier	Name	Type	Range	Access	Default	Mandatory /Optional
0x0400	NoTierBlock1Price	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x0401	NoTierBlock2Price	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x0402	NoTierBlock3Price	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x040N	NoTierBlockN+1Price ...	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x040F	NoTierBlock16Price	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x0410	Tier1Block1Price	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x0411	Tier1Block2Price	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x0412	Tier1Block3Price	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x041N	Tier1BlockN+1Price ...	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x041F	Tier1Block16Price	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x0420	Tier2Block1Price	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x042N	Tier2BlockN+1Price ...	Unsigned 32-bit integer	0x00000000 to 0xFFFFFFFF	Read only	-	O
0x042F	Tier2Block16Price	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

8060

8061

8062 D.4.2.2.5.1 TierNBlockNPrice Attributes

8063 Attributes *PriceNoTierBlock1* through *PriceTier15Block16* represent the price of Energy, Gas,
 8064 or Water delivered to the premises (i.e. delivered to the customer from the utility) at a specific
 8065 price tier as defined by a TOU schedule, Block Threshold or a real time pricing period. If
 8066 optionally provided, attributes shall be initialized prior to the issuance of associated *Publish*
 8067 *Price* commands (see sub-clause D.4.2.4.1). The expected practical limit for the number of
 8068 *PriceTierNBlockN* attributes supported is 32. The Unit of Measure, Currency and Trailing Digits
 8069 that apply to this attribute should be obtained from the appropriate fields in a Publish Price
 8070 command.

8071 D.4.2.2.6 **Extended Price Information (Delivered) Set**

8072 In case of TOU charging only, the price server allows support for up to 48 TOU rates. To reduce
 8073 the number of attributes, *Tier1Block1Price* through *Tier15Block1Price* attributes are reused to
 8074 represent rates for tiers 1 to 15. Rates for tiers 16 to 48 are provided in the extended price
 8075 information set.

8076 **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

8077

8078 D.4.2.2.6.1 **PriceTierN Attributes**

8079 Attributes *PriceTier16* through *PriceTier48* represent the price of Energy, Gas, or Water
 8080 delivered to the premises (i.e. delivered to the customer from the utility) at a specific price tier.

8081 D.4.2.2.6.2 **(CPP1 Price Attribute)**

8082 Attribute *CPP1 Price* represents the price of Energy, Gas, or Water delivered to the premises (i.e.
 8083 delivered to the customer from the utility) while Critical Peak Pricing ‘CPP1’ is being applied.

8084 D.4.2.2.6.3 **(CPP2 Price Attribute)**

8085 Attribute *CPP2 Price* represents the price of Energy, Gas, or Water delivered to the premises (i.e.
 8086 delivered to the customer from the utility) while Critical Peak Pricing ‘CPP2’ is being applied.

8087

8088 D.4.2.2.7 **Tariff Information (Delivered) Attribute Set**

8089 The following set of attributes represents items that are associated with a particular Price Tariff.
 8090 Please note that the terms tier and rate are used interchangeably here, but do define the same
 8091 thing.

8092 **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					

8093 D.4.2.2.7.1 TariffLabel Attribute

8094 The *TariffLabel* attribute provides a method for utilities to assign a label to an entire set of tariff
 8095 information. The *TariffLabel* attribute is a ZCL Octet String capable of storing a 24 character
 8096 string (the first Octet indicates length) encoded in the UTF-8 format. This attribute is thought of
 8097 be useful when a commodity supplier may have multiple tariffs. The *TariffLabel* attribute
 8098 represents the current active value.

8099 D.4.2.2.7.2 NumberofPriceTiersInUse Attribute

8100 An 8-bit integer which indicates the number of price tiers used while this tariff is active. Valid
 8101 values are from 0 to 48 reflecting block charging only (no price tiers in use) (0) to 48 price tiers
 8102 available (48). The *NumberofPriceTiersinUse* attribute represents the current active value.

8103 D.4.2.2.7.3 NumberofBlockThresholdsInUse Attribute

8104 An 8-bit integer which indicates the total number of block thresholds used in the currently active
 8105 tariff.



8106 When utilizing TOU charging only, the attribute shall be set to 0 (no thresholds employed).

8107 Where a single set of thresholds is employed, valid values are from 1 to 15 reflecting 1 to 15
 8108 block thresholds available. The number of blocks is one greater than the number of block
 8109 thresholds.

8110 Where the *TierBlockMode* is set to 2, this attribute indicates the sum of all thresholds employed
 8111 for all tiers within the currently active tariff.

8112 D.4.2.2.7.4 TierBlockMode Attribute

8113 An 8-bit enumeration indicating how the mixed TOU / Block charging is to be applied. The value
 8114 stored in this attribute is applicable only in the case where *NumberofPriceTiersInUse* is greater
 8115 than one and *NumberofBlockThresholdsInUse* is greater than zero. The following table shows
 8116 possible values.

8117

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-55.
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-56.
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-56 with the exception that the thresholds used can vary from tier to tier.
0x03 to 0xFE	Reserved for future use
0xFF	Not Used

8118

8119

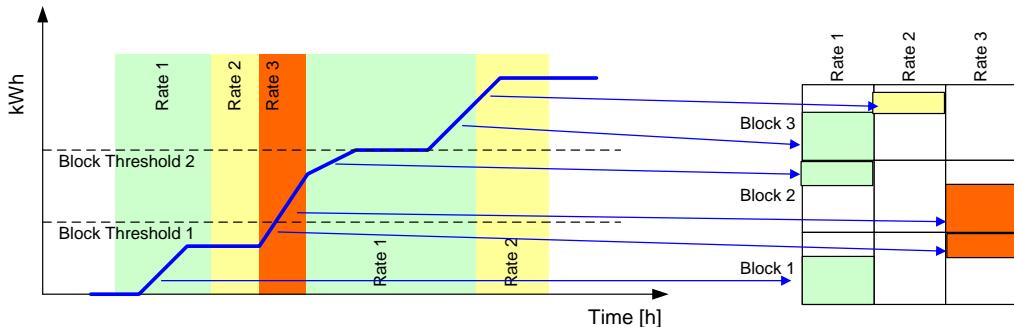
8120
8121

Figure D-55– Single Threshold Set applied to All Consumption

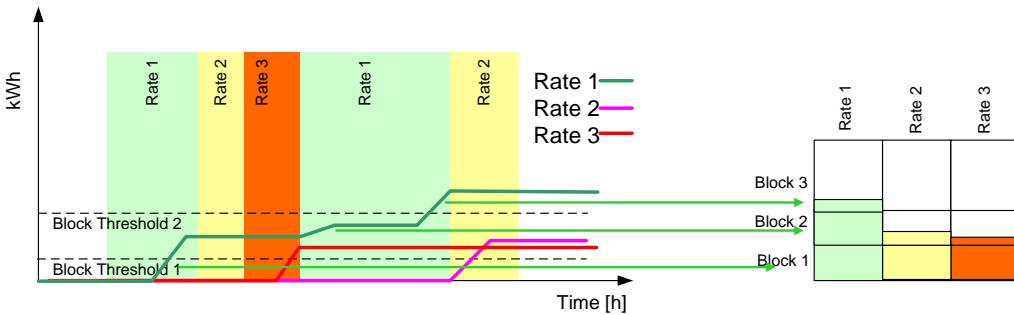
8122
81238124
8125

Figure D-56– Single Threshold Set applied to Each Tier Consumption

8126
8127
8128
8129

Note: Tiers 1-15 ONLY are available for hybrid Tier/Block tariffing ... Tiers 16-48 are for TOU tariffing only.

8130

D.4.2.2.7.5 Unit of Measure Attribute

An 8-bit enumeration identifying the base unit of measure. The enumeration used for this attribute shall match one of the *UnitOfMeasure* values using a pure Binary format, as defined in the Metering cluster.

8134

D.4.2.2.7.6 Currency Attribute

An unsigned 16-bit integer containing identifying information concerning the local unit of currency used in the Price cluster. The *Currency* attribute shall correspond to the *Currency* field within the *PublishPrice* command.

8138

The value of the currency attribute should match the values defined by ISO 4217.

8139 D.4.2.2.7.7 PriceTrailingDigit Attribute

8140 An 8-bit BitMap used to determine where the decimal point is located for prices provided in the
 8141 *Standing Charge* attribute and the *Price Matrix* command. The most significant nibble is the
 8142 Trailing Digit sub-field which indicates the number of digits to the right of the decimal point.
 8143 The least significant nibble is reserved and shall be 0. The *Price Trailing Digit* attribute
 8144 represents the current active value.

8145 D.4.2.2.7.8 TariffResolutionPeriod Attribute

8146 An 8 bit enumeration identifying the resolution period for Block Tariff, Table D-107 shows all
 8147 available options.

8148 D.4.2.2.7.9 CO₂ Attribute

8149 Used to calculate the amount of carbon dioxide (CO₂) produced from energy use. Natural gas has
 8150 a conversion factor of about 0.185, e.g. 1,000 kWh of gas used is responsible for the production
 8151 of 185kg CO₂ (0.185 x 1000 kWh). The CO₂ attribute represents the current active value.

8152 D.4.2.2.7.10 CO₂Unit Attribute

8153 This attribute is an 8-bit enumeration which defines the unit for the CO₂ attribute. The values and
 8154 descriptions for this attribute are listed in Table D-83 below. The CO₂Unit attribute represents
 8155 the current active value.

8156 **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

8157

8158 D.4.2.2.7.11 CO₂TrailingDigit Attribute

8159 An 8-bit Bit-Map used to determine where the decimal point is located in the CO₂ attribute. The
 8160 most significant nibble indicates the number of digits to the right of the decimal point. The least
 8161 significant nibble is reserved. The CO₂TrailingDigit attribute represents the current active value.

8162 D.4.2.2.8 Billing Information (Delivered) Attribute Set

8163 The following set of attributes provides remote access to the Price server Billing information.

8164 *Note:* The Billing Information Attribute Set in this revision of this specification is provisional
 8165 and not certifiable. This feature set may change before reaching certifiable status in a future
 8166 revision of this specification.

8167

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					

8168

8169 D.4.2.2.8.1 *CurrentBillingPeriodStart* Attribute8170 The *CurrentBillingPeriodStart* attribute represents the start time of the current billing period.8171 D.4.2.2.8.2 *CurrentBillingPeriodDuration* Attribute8172 The *CurrentBillingPeriodDuration* attribute represents the current billing period duration in
8173 minutes.8174 D.4.2.2.8.3 *LastBillingPeriodStart* Attribute8175 The *LastBillingPeriodStart* attribute represents the start time of the last billing period.8176 D.4.2.2.8.4 *LastBillingPeriodDuration* Attribute8177 The *LastBillingPeriodDuration* attribute is the duration of the last billing period in minutes (start
8178 to end of last billing period).8179 D.4.2.2.8.5 *LastBillingPeriodConsolidatedBill* Attribute8180 The *LastBillingPeriodConsolidatedBill* attribute is an amount for the cost of the energy supplied
8181 from the date of the *LastBillingPeriodStart* attribute and until the duration of the
8182 *LastBillingPeriodDuration* attribute expires, measured in base unit of *Currency* with the decimal
8183 point located as indicated by the Trailing Digits attribute.

8184 D.4.2.2.9 Credit Payment Attribute Set

8185 The Credit Payments Attribute set provides a method for the HAN (IHD) to understand the
 8186 current status of the credit-only payment made to the energy supplier. These payments are for a
 8187 credit meter only and do not cover any Prepayment Top up or payment. This attribute set is used
 8188 to display the bill on the IHD should this service be required. Devices that require this
 8189 information should use standard ZCL commands to read this information.

8190 **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					
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

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
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					

8191

8192 D.4.2.2.9.1 CreditPaymentdueDate Attribute8193 The *CreditPaymentDueDate* attribute indicates the date and time when the next credit payment is
8194 due to be paid by the consumer to the supplier.8195 D.4.2.2.9.2 CreditPaymentStatus Attribute8196 The *CreditPaymentStatus* attribute indicates the current status of the last payment. Table D-86
8197 defines the enumeration values for this attribute.

8198

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

8199

8200 D.4.2.2.9.3 CreditPaymentOverDueAmount Attribute

8201 This is the total of the consolidated bill amounts accumulated since the last payment.

8202 D.4.2.2.9.4 PaymentDiscount Attribute8203 The *PaymentDiscount* attribute indicates the discount that the energy supplier has applied to the
8204 consolidated bill.8205 D.4.2.2.9.5 PaymentDiscountPeriod Attribute8206 The *PaymentDiscountPeriod* attribute indicates the period for which this discount shall be
8207 applied for. Table D-87 shows the enumeration values for this attribute.

8208

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

8209

8210 D.4.2.2.9.6 CreditPayment Attribute

8211 The *CreditPayment* attributes indicate the amount paid by the consumer to the energy supplier.
 8212 The last 5 values are shown with #1 meaning the most recent. Measured in base unit of *Currency*
 8213 with the decimal point located as indicated by the *Trailing Digits* attribute.

8214 D.4.2.2.9.7 CreditPaymentDate Attribute

8215 The *CreditPaymentDate* attributes indicate the last time the consumer made a payment to the
 8216 energy supplier. The last 5 values are shown with #1 meaning the most recent.

8217 D.4.2.2.9.8 CreditPaymentRef Attribute

8218 The *CreditPaymentRef* attributes indicate the reference number given to the payment by the
 8219 energy supplier. The last 5 values are shown with #1 meaning the most recent.

8220 D.4.2.2.10 Received Tier Label Attribute Set8221
8222

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					

8223

8224 D.4.2.2.10.1 ReceivedTierNPriceLabel Attributes

8225 The *ReceivedTierNPriceLabel* attributes provide a method for utilities to assign a label to
 8226 Received Price Tiers. There are 48 Tier Labels. The format and use of these attributes is the same
 8227 as for the ‘Delivered’ Price Labels defined in D.4.2.2.1.1.

8228 D.4.2.2.11 **Received Block Threshold Attribute Set**

8229 The following set of attributes provides remote access to the Price server
 8230 *ReceivedBlockThresholds*. The number of block thresholds is indicated by the
 8231 *NumberofBlockThresholds* field in the associated *PublishTariffInformation* command. The
 8232 number of blocks is one greater than the number of thresholds.

8233 **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					

8234

8235 D.4.2.2.11.1 **ReceivedBlockNThreshold Attributes**

8236 The format of these attributes is the same as for the ‘Delivered’ Block Thresholds defined in
 8237 D.4.2.2.2.1.

8238 D.4.2.2.12 **Received Block Period Attribute Set**

8239 The following set of attributes provides remote access to the Price server Received Block
 8240 Threshold period (typically the billing cycle) information.

8241 **Table D-90– Received Block Period Attribute Set**

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x8200	ReceivedStartofBlockPeriod	UTC Time				
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					

8242

8243 D.4.2.2.12.1 ReceivedStartofBlockPeriod Attribute

8244 The format of this attribute is the same as for the ‘Delivered’ *StartofBlockPeriod* attribute
 8245 defined in D.4.2.2.3.1.

8246 D.4.2.2.12.2 ReceivedBlockPeriodDuration Attribute

8247 The format of this attribute is the same as for the ‘Delivered’ *BlockPeriodDuration* attribute
 8248 defined in D.4.2.2.3.2.

8249 D.4.2.2.12.3 ReceivedThresholdMultiplier Attribute

8250 The format of this attribute is the same as for the ‘Delivered’ *ThresholdMultiplier* attribute
 8251 defined in D.4.2.2.3.3.

8252 D.4.2.2.12.4ReceivedThresholdDivisor Attribute

8253 The format of this attribute is the same as for the ‘Delivered’ *ThresholdDivisor* attribute
 8254 defined in D.4.2.2.3.4.

8255 D.4.2.2.13 Received Block Price Information Attribute Set

8256 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

8257

8258 D.4.2.2.13.1 RxTierNBlockNPrice Attributes

8259 The format and use of these attributes is the same as for the ‘Delivered’ *TierNBlockNPrice*
 8260 attributes defined in D.4.2.2.5.1.

8261 D.4.2.2.14 Received Extended Price Information Attribute Set

8262 In case of TOU charging only, the price server shall support up to 48 TOU rates. To reduce the
 8263 number of attributes, *RxTierNBlock1Price* attributes are reused to represent rates for tiers 1 to 15.
 8264 Rates for tiers 16 to 48 are provided in the Received Extended Price Information Set.

8265 **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					

8266 D.4.2.2.14.1 ReceivedPriceTierN Attributes

8267 The format and use of these attributes is the same as for the ‘Delivered’ *PriceTierN* attributes
 8268 defined in D.4.2.2.6.1.

8269 D.4.2.2.15 Received Tariff Information Attribute Set

8270 The following set of attributes represents items that are associated with a particular Received
 8271 Price Tariff.

8272 **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	ReceivedNumberofPriceTiersInUse	Unsigned 8-bit Integer	0 to 15	Read Only	0	O

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x8612	ReceivedNumberofBlockThresholdsInUse	Unsigned 8-bit Integer	0 to 15	Read Only	0	O
0x8613	ReceivedTierBlockMode	Unsigned 8-bit Integer	0 to 1	Read Only	0xFF	O
0x8614	Reserved					
0x8615	ReceivedTariffResolutionPeriod	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					

8273

8274 D.4.2.2.15.1 ReceivedTariffLabel Attribute8275 The format and use of this attribute is the same as for the ‘Delivered’ *TariffLabel* attribute
8276 defined in D.4.2.2.7.1.8277 D.4.2.2.15.2 ReceivedNumberofPriceTiersInUse Attribute8278 The format and use of this attribute is the same as for the ‘Delivered’ *NumberofPriceTiersInUse*
8279 attribute defined in D.4.2.2.7.2.8280 D.4.2.2.15.3 ReceivedNumberofBlockThresholdsInUse Attribute8281 The format and use of this attribute is the same as for the ‘Delivered’
8282 *NumberofBlockThresholdsInUse* attribute defined in D.4.2.2.7.3.8283 D.4.2.2.15.4 ReceivedTierBlockMode Attribute8284 The format and use of this attribute is the same as for the ‘Delivered’ *TierBlockMode* attribute
8285 defined in D.4.2.2.7.4.8286 D.4.2.2.15.5 ReceivedTariffResolutionPeriod Attribute8287 An 8 bit enumeration identifying the resolution period for Block Tariff, Table D-107 shows all
8288 available options.

8289 D.4.2.2.15.6 ReceivedCO₂ Attribute

8290 The format and use of this attribute is the same as for the ‘Delivered’ *CO₂* attribute defined in
8291 D.4.2.2.7.9.

8292 D.4.2.2.15.7 ReceivedCO₂Unit Attribute

8293 The format and use of this attribute is the same as for the ‘Delivered’ *CO₂Unit* attribute defined
8294 in D.4.2.2.7.10.

8295 D.4.2.2.15.8 ReceivedCO₂TrailingDigit Attribute

8296 The format and use of this attribute is the same as for the ‘Delivered’ *CO₂TrailingDigit* attribute
8297 defined in D.4.2.2.7.11.

8298 D.4.2.2.16 **Received Billing Information Attribute Set**

8299 The following set of attributes represents items that are associated with particular Received
8300 Billing information.

8301

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
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					

8302

8303 D.4.2.2.16.1 ReceivedCurrentBillingPeriodStart Attribute

8304 The format and use of this attribute is the same as for the ‘Delivered’ *CurrentBillingPeriodStart*
8305 attribute defined in D.4.2.2.8.1.

8306 D.4.2.2.16.2 ReceivedCurrentBillingPeriodDuration Attribute

8307 The format and use of this attribute is the same as for the ‘Delivered’
8308 *CurrentBillingPeriodDuration* attribute defined in D.4.2.2.8.2.

8309 D.4.2.2.16.3 ReceivedLastBillingPeriodStart Attribute

8310 The format and use of this attribute is the same as for the ‘Delivered’ *LastBillingPeriodStart*
 8311 attribute defined in D.4.2.2.8.3.

8312 D.4.2.2.16.4 ReceivedLastBillingPeriodDuration Attribute

8313 The format and use of this attribute is the same as for the ‘Delivered’ *LastBillingPeriodDuration*
 8314 attribute defined in D.4.2.2.8.4.

8315 D.4.2.2.16.5 ReceivedLastBillingPeriodConsolidatedBill Attribute

8316 The format and use of this attribute is the same as for the ‘Delivered’
 8317 *LastBillingPeriodConsolidatedBill* attribute defined in D.4.2.2.8.5.

8318

8319 D.4.2.3 Commands Received

8320 The server side of the Price cluster is capable of receiving the commands listed in Table D-95.

8321 **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	

8322

8323 D.4.2.3.1 **Error Handling**

8324 If the response to a ‘Get’ command has no data available, then the device should respond using a
 8325 ZCL Default Response with a status of NOT_FOUND.

8326 D.4.2.3.2 **GetCurrentPrice Command** 

8327 This command initiates a *PublishPrice* command (see sub-clause D.4.2.4.1) for the current time.

8328 D.4.2.3.2.1 **Payload Format**

8329 The payload of the *GetCurrentPrice* command is formatted as shown in Figure D-57:

Octets	1
Data Type	Unsigned 8-bit integer
Field Name	Command Options

8330 **Figure D-57– *GetCurrentPrice* Command Payload**

8331

8332 **D.4.2.3.2.1.1 Payload Details**

8333 **The Command Options Field:** The command options field is 8 Bits in length and is formatted as
 8334 a bit field as shown in Figure D-58.

8335

Bits	0	1 to 7
Field Name	Requestor Rx On When Idle	Reserved

8336 **Figure D-58– *Get Current Price* Command Options Field**

8337 **The Requestor Rx On When Idle Sub-field:** The Requestor Rx On When Idle sub-field has a
 8338 value of 1 if the requestor’s receiver may be, for all practical purposes, enabled when the
 8339 device is not actively transmitting, thereby making it very likely that regular broadcasts of
 8340 pricing information will be received by this device, and 0 otherwise.

8341 A device that publishes price information may use the value of this bit, as received from requestors
 8342 in its neighborhood, to determine publishing policy. For example, if a device makes a request for
 8343 current pricing information and the requestor Rx on when idle sub-field of the *GetCurrentPrice*
 8344 command payload has a value of 1 (indicating that the device will be likely to receive regular
 8345 price messages), then the receiving device may store information about the requestor and use it in
 8346 future publishing operations.

8347 D.4.2.3.2.2 Effect on Receipt

8348 On receipt of this command, the device shall send a *PublishPrice* command (sub-clause
8349 D.4.2.4.1) for the currently scheduled time.

8350

8351 D.4.2.3.3 GetScheduledPrices Command

8352 This command initiates a *PublishPrice* command (see sub-clause D.4.2.4.1) for available price
8353 events. A server device shall be capable of storing five price events at a minimum.

8354 D.4.2.3.3.1 Payload Details

8355 The *GetScheduledPrices* command payload shall be formatted as illustrated in Figure D-59:

8356

Octets	4	1
Data Type	UTCTime	Unsigned8-bit integer
Field Name	StartTime (M)	NumberofEvents (M)

8357 **Figure D-59—GetScheduledPrices Command Payload**

8358 **StartTime (mandatory):** UTC Timestamp representing the minimum ending time for any
8359 scheduled or currently active pricing events to be resent. If a command has a *StartTime* of
8360 0x00000000, replace that *StartTime* with the current time stamp.

8361 **NumberofEvents (mandatory):** Represents the maximum number of events to be sent. A value
8362 of 0 would indicate all available events are to be returned. Example: Number of Events = 1
8363 would return the first event with an EndTime greater than or equal to the value of *StartTime*
8364 field in the *GetScheduledPrices* command. (EndTime would be *StartTime* plus Duration of the
8365 event listed in the device's event table).

8366 D.4.2.3.3.2 When Generated

8367 This command is generated when the client device wishes to verify the available Price Events or
8368 after a loss of power/reset occurs and the client device needs to recover currently active,
8369 scheduled, or expired Price Events.

8370 A ZCL Default Response with status NOT_FOUND shall be returned if there are no events
8371 available.

8372 D.4.2.3.3.3 Effect on Receipt

8373 On receipt of this command, the device shall send a *PublishPrice* command (see sub-clause
8374 D.4.2.4.1) for all currently scheduled price events.

8375 D.4.2.3.4 **PriceAcknowledgement Command**

8376 The *PriceAcknowledgement* command described in Figure D-60 provides the ability to
 8377 acknowledge a previously sent *PublishPrice* command. It is mandatory for 1.1 and later devices.
 8378 For SE 1.0 devices, the command is optional.

8379 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)

8380 **Figure D-60—PriceAcknowledgement Command Payload**8381 D.4.2.3.4.1.1 **Payload Details**

8382 **Provider ID (mandatory):** An unsigned 32 bit field containing a unique identifier for
 8383 the commodity provider.

8384 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider.

8385 **Price Ack Time (mandatory):** Time price acknowledgement generated.

8386 **Control (mandatory):** Identifies the Price Control or Block Period Control options for the
 8387 event. The values for this field are described in Table D-100 and Figure D-74.

8388 D.4.2.3.4.2 **When Generated**

8389 This command is generated on receipt of a *PublishPrice* command when the PriceControl
 8390 field of that *PublishPrice* command indicates that a Price Acknowledgement is required
 8391 (see sub-clause D.4.2.4.1 for further details).

8392

8393 D.4.2.3.5 **GetBlockPeriod(s) Command**

8394 This command initiates a *PublishBlockPeriod* command (see sub-clause D.4.2.4.2) for the
 8395 currently scheduled block periods. A server device shall be capable of storing at least two
 8396 commands, the current period and a period to be activated in the near future.

8397 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)

8398

Figure D-61– Get Block Period(s) Command Payload

8399

D.4.2.3.5.1.1 Payload Details8400
8401
8402

Start Time (mandatory): UTC Timestamp representing the minimum ending time for any scheduled or currently block period events to be resent. If a command has a Start Time of 0x00000000, replace that Start Time with the current time stamp.

8403
8404
8405
8406
8407
8408

Number of Events (mandatory): An 8 bit Integer which indicates the maximum number of *Publish Block Period* commands that can be sent. Example: Number of Events = 1 would return the first event with an EndTime greater than or equal to the value of Start Time field in the *GetBlockPeriod(s)* command. (EndTime would be StartTime plus Duration of the event listed in the device's event table). Number of Events = 0 would return all available Publish Block Periods, starting with the current block in progress.

8409
8410
8411
8412
8413

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.). If the *TariffType* is not specified, the server shall assume that the request is for the 'Delivered' Tariff. The most significant nibble is reserved.

8414

D.4.2.3.5.2 When Generated8415
8416
8417

This command is generated when the client device wishes to verify the available Block Period events or after a loss of power/reset occurs and the client device needs to recover currently active or scheduled Block Periods.

8418
8419

A ZCL Default response with status NOT_FOUND shall be returned if there are no events available.

8420

D.4.2.3.5.3 Effect on Receipt8421
8422
8423

On receipt of this command, the device shall send a *Publish Block Period* command (sub-clause D.4.2.4.2) for all currently scheduled periods, up to the maximum number of commands specified.

8424

8425

D.4.2.3.6 GetConversionFactor Command8426
8427
8428

This command initiates a *PublishConversionFactor* command(s) for scheduled conversion factor updates. A server device shall be capable of storing at least two instances, the current and (if available) next instance to be activated in the future.

8429

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)

8430

Figure D-62– GetConversionFactor Command Payload8431 D.4.2.3.6.2 Payload Details

8432 **Earliest Start Time (mandatory):** UTC Timestamp indicating the earliest start time of values to
 8433 be returned by the corresponding *PublishConversionFactor* command. The first returned
 8434 *PublishConversionFactor* command shall be the instance which is active or becomes active at
 8435 or after the stated *Earliest Start Time*. If more than one instance is requested, the active and
 8436 scheduled instances shall be sent with ascending ordered StartTime.

8437 **Min. Issuer Event ID (mandatory):** A 32-bit integer representing the minimum *Issuer Event ID*
 8438 of values to be returned by the corresponding *PublishCalorificValue* command. A value of
 8439 0xFFFFFFFF means not specified; the server shall return values irrespective of the value of the
 8440 *Issuer Event ID*.

8441 **Number of Commands (mandatory):** An 8-bit integer which represents the maximum number
 8442 of *PublishConversionFactor* commands that the client is willing to receive in response to this
 8443 command. A value of 0 would indicate all available *PublishConversionFactor* commands shall
 8444 be returned.

8445

8446 D.4.2.3.7 GetCalorificValue Command

8447 This command initiates a *PublishCalorificValue* command(s) for scheduled calorific value
 8448 updates. A server device shall be capable of storing at least two instances, the current and (if
 8449 available) next instance to be activated in the future.

8450 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)

8451

Figure D-63– GetCalorificValue Command Payload8452 D.4.2.3.7.2 Payload Details

8453 **Earliest Start Time (mandatory):** UTC Timestamp indicating the earliest start time of values to
 8454 be returned by the corresponding *PublishCalorificValue* command. The first returned
 8455 *PublishCalorificValue* command shall be the instance which is active or becomes active at or

8456 after the stated *Earliest Start Time*. If more than one instance is requested, the active and
 8457 scheduled instances shall be sent with ascending ordered Start Time.

8458 **Min. Issuer Event ID (mandatory):** A 32-bit integer representing the minimum *Issuer Event ID*
 8459 of values to be returned by the corresponding *PublishCalorificValue* command. A value of
 8460 0xFFFFFFFF means not specified; the server shall return values irrespective of the value of the
 8461 *Issuer Event ID*.

8462 **Number of Commands (mandatory):** An 8-bit Integer which represents the maximum number
 8463 of *PublishCalorificValue* commands that the client is willing to receive in response to this
 8464 command. A value of 0 would indicate all available *PublishCalorificValue* commands shall be
 8465 returned.

8466

8467 **D.4.2.3.8 GetTariffInformation Command**

8468 This command initiates *PublishTariffInformation* command(s) for scheduled tariff updates. A
 8469 server device shall be capable of storing at least **two** instances, current and the next instance to be
 8470 activated in the future.

8471 One or more *PublishTariffInformation* commands are sent in response to this command.

8472 To obtain the complete tariff details, further *GetPriceMatrix* and *GetBlockThresholds* commands
 8473 must be sent using the start time and *IssuerTariffID* obtained from the appropriate
 8474 *PublishTariffInformation* command.

8475 **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)

8476 **Figure D-64– GetTariffInformation Command Payload**

8477

8478 **D.4.2.3.8.2 Payload Details**

8479 **Earliest Start Time (mandatory):** UTC Timestamp indicating the earliest start time of tariffs to
 8480 be returned by the corresponding *PublishTariffInformation* command. The first returned
 8481 *PublishTariffInformation* command shall be the instance which is active or becomes active at or
 8482 after the stated *EarliestStartTime*. If more than one command is requested, the active and
 8483 scheduled commands shall be sent with ascending ordered *StartTime*.

8484 **Min. Issuer Event ID (mandatory):** A 32-bit integer representing the minimum *Issuer Event ID*
8485 of tariffs to be returned by the corresponding *PublishTariffInformation* command. A value of
8486 0xFFFFFFFF means not specified; the server shall return tariffs irrespective of the value of the
8487 *Issuer Event ID*.

8488 **Number of Commands (mandatory):** An 8-bit integer which represents the maximum number
8489 of *PublishTariffInformation* commands that the client is willing to receive in response to this
8490 command. A value of 0 would indicate all available *PublishTariffInformation* commands shall be
8491 returned.

8492 **Tariff Type (mandatory):** An 8-bit bitmap identifying the type of tariff published in this
8493 command. The least significant nibble represents an enumeration of the tariff type as detailed in
8494 Table D-108 (Generation Meters shall use the ‘Received’ Tariff.). The most significant nibble is
8495 reserved.

8496

8497 D.4.2.3.9 **GetPriceMatrix Command**

8498 This command initiates a *PublishPriceMatrix* command for the scheduled Price Matrix updates.
8499 A server device shall be capable of storing at least **two** instances, current and next instance to be
8500 activated in the future.

8501 D.4.2.3.9.1 **Payload Format**

Octets	4
Data Type	Unsigned 32-bit Integer
Field Name	Issuer Tariff ID (M)

8502 **Figure D-65– GetPriceMatrix Command Payload**

8503 D.4.2.3.9.2 **Payload Details**

8504 **Issuer Tariff ID (mandatory):** *IssuerTariffID* indicates the tariff to which the requested Price
8505 Matrix belongs.

8506 **Note:** A Price Matrix instance may require multiple *PublishPriceMatrix* commands to be
8507 transmitted to the client device.

8508

8509 D.4.2.3.10 **GetBlockThresholds Command**

8510 This command initiates a *PublishBlockThreshold* command for the scheduled Block Threshold
8511 updates. A server device shall be capable of storing at least **two** instances, current and next
8512 instance to be activated in the future.

8513 D.4.2.3.10.1 Payload Format

Octets	4
Data Type	Unsigned 32-bit Integer
Field Name	Issuer Tariff ID (M)

8514 **Figure D-66– GetBlockThresholds Command Payload**

8515 D.4.2.3.10.2 Payload Details

8516 **Issuer Tariff ID (mandatory):** Issuer Tariff ID indicates the tariff to which the requested Block
8517 Thresholds belong.

8518 **Note:** A Block Threshold instance may require multiple *PublishBlockThreshold* commands to be
8519 transmitted to the client device.

8520

8521 D.4.2.3.11 GetCO₂Value Command

8522 This command initiates *PublishCO₂Value* command(s) for scheduled CO₂ conversion factor
8523 updates. A server device shall be capable of storing at least **two** instances, current and (if
8524 available) next instance to be activated in the future.

8525 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)

8526 **Figure D-67– GetCO₂Value Command Payload**

8527 D.4.2.3.11.2 Payload Details

8528 **Earliest Start Time (mandatory):** UTC Timestamp indicating the earliest start time of values to
8529 be returned by the corresponding *PublishCO₂Value* command. The first returned
8530 *PublishCO₂Value* command shall be the instance which is active or becomes active at or after the
8531 stated *EarliestStartTime*. If more than one instance is requested, the active and scheduled
8532 instances shall be sent with ascending ordered *StartTime*.

8533 **Min. Issuer Event ID (mandatory):** A 32-bit integer representing the minimum *Issuer Event ID*
8534 of values to be returned by the corresponding *PublishCO₂Value* command. A value of
8535 0xFFFFFFFF means not specified; the server shall return values irrespective of the value of the
8536 *Issuer Event ID*.

8537 **Number of Commands (mandatory):** An 8-bit Integer which represents the maximum number
8538 of *PublishCO₂Value* commands that the client is willing to receive in response to this command.
8539 A value of 0 would indicate all available *PublishCO₂Value* commands shall be returned.

8540 **Tariff Type (Optional):** An 8-bit bitmap identifying the type of tariff published in this
 8541 command. The least significant nibble represents an enumeration of the tariff type as detailed in
 8542 Table D-108 (Generation Meters shall use the ‘Received’ Tariff). A value of 0xFF means not
 8543 specified. If the *TariffType* is not specified, the server shall return all C0₂ values regardless of
 8544 tariff type. The most significant nibble is reserved.

8545

8546 D.4.2.3.12 **GetTierLabels Command**

8547 This command allows a client to retrieve the tier labels associated with a given tariff; this
 8548 command initiates a *PublishTierLabels* command from the server.

8549 D.4.2.3.12.1 **Payload Format**

Octets	4
Data Type	Unsigned 32-bit Integer
Field Name	Issuer Tariff ID (M)

8550 **Figure D-68— GetTierLabels Command Payload**8551 D.4.2.3.12.2 **Payload Details**

8552 **Issuer Tariff ID (mandatory):** Unique identifier generated by the commodity supplier. This is
 8553 used to identify the tariff that the labels apply to.

8554

8555 D.4.2.3.13 **GetBillingPeriod Command**

8556 This command initiates one or more *PublishBillingPeriod* commands for currently scheduled
 8557 billing periods.

8558 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)

8559 **Figure D-69— GetBillingPeriod Command Payload**8560 D.4.2.3.13.2 **Payload Details**

8561 **Earliest Start Time (mandatory):** UTC Timestamp indicating the earliest start time of billing
 8562 periods to be returned by the corresponding *PublishBillingPeriod* command. The first returned
 8563 *PublishBillingPeriod* command shall be the instance which is active or becomes active at or after
 8564 the stated *EarliestStartTime*. If more than one instance is requested, the active and scheduled
 8565 instances shall be sent with ascending ordered *StartTime*.

Min. Issuer Event ID (mandatory): A 32-bit integer representing the minimum *Issuer Event ID* of billing periods to be returned by the corresponding *PublishBillingPeriod* command. A value of 0xFFFFFFFF means not specified; the server shall return periods irrespective of the value of the *Issuer Event ID*.

8570 **Number of Commands (mandatory):** An 8 bit Integer which indicates the maximum number of
8571 *PublishBillingPeriod* commands that the client is willing to receive in response to this command.
8572 A value of 0 would indicate all available *PublishBillingPeriod* commands shall be returned.

Tariff Type (optional): An 8-bit bitmap identifying the *TariffType* of the requested Billing Period information. 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 Billing Period information regardless of its type. The most significant nibble is reserved.

8578

8579 D.4.2.3.14 [GetConsolidatedBill Command](#)

8580 This command initiates one or more *PublishConsolidatedBill* commands with the requested
8581 billing information.

8582 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-70– GetConsolidatedBill Command Payload

8584 D.4.2.3.14.2 Payload Details

8585 **Earliest Start Time (mandatory):** UTC Timestamp indicating the earliest start time of billing
8586 information to be returned by the corresponding *PublishConsolidatedBill* command. The first
8587 returned *PublishConsolidatedBill* command shall be the instance which is active or becomes
8588 active at or after the stated *EarliestStartTime*. If more than one instance is requested, the active
8589 and scheduled instances shall be sent with ascending ordered *StartTime*.

Min. Issuer Event ID (mandatory): A 32-bit integer representing the minimum *Issuer Event ID* of 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*.

8594 **Number of Commands (mandatory):** An 8 bit Integer which indicates the maximum number of
8595 *PublishConsolidatedBill* commands that can be sent. A value of 0 would indicate all available
8596 *PublishConsolidatedBill* commands shall be returned.

8597 **Tariff Type (Optional):** An 8-bit bitmap identifying the type of tariff published in this
 8598 command. The least significant nibble represents an enumeration of the tariff type as detailed in
 8599 Table D-108 (Generation Meters shall use the ‘Received’ Tariff). A value of 0xFF means not
 8600 specified. If the *TariffType* is not specified, the server shall return all billing information
 8601 regardless of tariff type. The most significant nibble is reserved.

8602

8603 **D.4.2.3.15 CPPEventResponse Command**

8604 The *CPPEventResponse* command is sent from a Client (IHD) to the ESI to notify it of a Critical
 8605 Peak Pricing event authorization.

8606 **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)

8607 **Figure D-71– CPPEventResponse Command Payload**

8608 **D.4.2.3.15.2 Payload Details**

8609 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. When
 8610 new information is provided that replaces older information for the same time period, this field
 8611 allows devices to determine which information is newer. The value contained in this field is a
 8612 unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)
 8613 identifying when the Publish command was issued. Thus, newer information will have a value in
 8614 the *Issuer Event ID* field that is larger than older information.

8615 **CPP Auth (mandatory):** An 8-bit enumeration identifying the status of the CPP event. This
 8616 field shall contain the ‘Accepted’ or ‘Rejected’ values defined in Table D-112.

8617 **D.4.2.3.15.3 When Generated**

8618 The *CPPEventResponse* command is sent in response to the *PublishCPPEvent* command, for
 8619 either the Meter or the IHD, as acceptance or rejection of the CPP event.

8620 **D.4.2.3.15.4 Effect on Receipt**

8621 When the *CPPEventResponse* is received by the ESI, it will look at the *CPPAuth* parameter to
 8622 determine what action shall be taken next.

8623 The ESI shall resend the *PublishCPPEvent* command, but with the *CPPAuth* field now set to the
 8624 value received in the *CPPEventResponse* command.

8625

8626 D.4.2.3.16 **GetCreditPayment Command**

8627 This command initiates *PublishCreditPayment* commands for the requested credit payment
8628 information.

8629 D.4.2.3.16.1 **Payload Format**

8630 The *GetCreditPayment* command payload shall be formatted as illustrated in Figure D-72.

Octets	4	1
Data Type	UTCTime	Unsigned 8-bit Integer
Field Name	Latest End Time (M)	NumberOf Records (M)

8631 **Figure D-72– *GetCreditPayment* Command Payload**

8632 D.4.2.3.16.2 **Payload Details**

8633 **Latest End Time (mandatory):** UTC timestamp indicating the latest *CreditPaymentDate*
8634 of records to be returned by the corresponding *PublishCreditPayment* commands. The
8635 first returned *PublishCreditPayment* command shall be the most recent record with its
8636 *CreditPaymentDate* equal to or older than the *Latest End Time* provided.

8637 **NumberofRecords (mandatory):** An 8-bit integer that represents the maximum number of
8638 *PublishCreditPayment* commands that the client is willing to receive in response to this
8639 command. A value of 0 would indicate all available *PublishCreditPayment* commands shall be
8640 returned. If more than one record is requested, the *PublishCreditPayment* commands should be
8641 returned with descending ordered *CreditPaymentDate*. If fewer records are available than are
8642 being requested, only those available are returned.

8643

8644 D.4.2.3.17 **GetCurrencyConversion Command**

8645 This command initiates a *PublishCurrencyConversion* command for the currency conversion
8646 factor updates. A server shall be capable of storing both the old and the new currencies.

8647 D.4.2.3.17.1 **Payload Details**

8648 This command has no payload.

8649

8650 D.4.2.3.18 GetTariffCancellation Command

8651 This command initiates the return of the last *CancelTariff* command held on the associated
8652 server.

8653 D.4.2.3.18.1 Payload Details

8654 This command has no payload.

8655 D.4.2.3.18.2 When Generated

8656 This command is generated when the client device wishes to fetch any pending *CancelTariff*
8657 command from the server (see D.4.2.4.15 for further details). In the case of a BOMD, this may
8658 be as a result of the associated Notification flag.

8659 A ZCL Default response with status NOT_FOUND shall be returned if there is no *CancelTariff*
8660 command available.

8661

8662 D.4.2.4 Commands Generated

8663 The server side of the Price cluster is capable of generating the commands listed in Table D-96.

8664 **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	
-------------	----------	--

8665

8666 D.4.2.4.1 **Publish Price Command**

8667 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.

8672 When a *Get Current Price* or *Get Scheduled Prices* command is received over a ZigBee Smart Energy network, the *Publish Price* command should be sent unicast to the requester. In the case of an update to the pricing information from the commodity provider, the *Publish Price* command should be unicast to all individually registered devices implementing the Price Cluster on the ZigBee Smart Energy network.

8677 Devices capable of receiving this command must be capable of storing and supporting at least two pricing information instances, the current active price and the next price. By supporting at least two pricing information instances, receiving devices will allow the *Publish Price* command generator to publish the next pricing information during the current pricing period.

8681 Nested and overlapping *Publish Price* commands are not allowed. The current active price will be replaced if new price information is received by the ESI. In the case of overlapping events, the event with the newer Issuer Event ID takes priority over all nested and overlapping events. All existing events that overlap, even partially, should be removed. The only exception to this is that if an event with a newer Issuer Event ID overlaps with the end of the current active price but is not yet active, the active price is not deleted but its duration is modified to 0xFFFF (until changed) so that the active price ends when the new event begins.

8688 D.4.2.4.1.1 Payload Format

8689 The *PublishPrice* command payload shall be formatted as illustrated in Figure D-73.

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)

8690

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)

8691

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)

8692

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)

8693

Figure D-73– Publish Price Command Payload

8694 **Note:** M = Mandatory field, O = Optional field. **All fields must be present in the payload.**
 8695 **Optional fields will be marked with specific values to indicate they are not being used.**

8696 **Provider ID (mandatory):** An unsigned 32-bit field containing a unique identifier for
 8697 the commodity provider. This field allows differentiation in deregulated markets where multiple
 8698 commodity providers may be available.

8699 **Rate Label (mandatory):** A ZCL Octet String field capable of storing a 12 character string
 8700 (**the first Octet indicates length**) containing commodity provider- specific information regarding
 8701 the current billing rate. The String shall be encoded in the UTF-8 format. This field allows
 8702 differentiation when a commodity provider may have multiple pricing plans.

8703 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. When
 8704 new pricing information is provided that replaces older pricing information for the same time
 8705 period, **this field allows devices to determine which information is newer.** It is expected that the
 8706 **value contained in this field is a unique number managed by upstream servers or a UTC**
 8707 **based time stamp** (UTCTime data type) identifying when the *Publish Price* command was

8708 issued. Thus, newer pricing information will have a value in the Issuer Event ID field that is
 8709 larger than older pricing information.

8710 **Current Time (mandatory):** A UTCTime field containing the current time as determined by
 8711 the device. This field provides an extra value-added feature for the broadcast price signals.

8712 **Unit of Measure (mandatory):** An 8-bit enumeration field identifying the commodity as well as
 8713 its base unit of measure. The enumeration used for this field shall match one of the
 8714 UnitOfMeasure values using a pure binary format as defined in the Metering cluster (see sub-
 8715 clause D.3.2.2.4.1) 

8716 **Currency (mandatory):** An unsigned 16-bit field containing identifying information concerning
 8717 the local unit of currency used in the price field. This field allows the displaying of the appropriate
 8718 symbol for a currency (i.e.: \$). 

8719 The value of the currency field should match the values defined by ISO 4217.

8720 **Price Trailing Digit and Price Tier (mandatory):** An 8-bit field used to determine where the
 8721 decimal point is located in the price field and to indicate the current pricing tier as chosen by the
 8722 commodity provider. The most significant nibble is the Trailing Digit sub-field which indicates the
 8723 number of digits to the right of the decimal point. The least significant nibble is an enumerated field
 8724 containing the current Price Tier.

8725 Valid values for the Price Tier sub-field are from 1 to 15 reflecting the least expensive tier (1) to the
 8726 most expensive tiers (15). A value of zero indicates no price tier is in use. This parameter also
 8727 references the associated *TiernPriceLabel* attribute assigned to the Price Tier. Table D-97 depicts
 8728 the assignments. The meaning of value 0xF is dependant on the value of the optional *Extended*
 8729 *Price Tier* field. Absence of this field, or a value of 0x00 in this field, indicates that the current
 8730 Price Tier is fifteen, and references the *Tier15PriceLabel* attribute.. Where the *Extended Price*
 8731 *Tier* field contains a non-zero value, the current Price Tier and *TiernPriceLabel* attribute are
 8732 determined by the sum of the values of the *Price Tier* sub-field and the *Extended Price Tier* field.

8733 **Note:** Values for Price Tier listed above 0x6 in this revision of this specification are provisional
 8734 and not certifiable. This number of fields may change before reaching certifiable status in a
 8735 future revision of this specification.

8736 **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

8737

8738 **Number of Price Tiers & Register Tier (mandatory):** An 8-bit BitMap where the most
 8739 significant nibble is an enumerated sub-field representing the maximum number of price tiers
 8740 available, and the least significant nibble is an enumerated sub-field indicating the register tier
 8741 used with the current Price Tier.

8742 Valid values for the Number of Price Tiers sub-field are from 0 to 15 reflecting no tiers in use (0)
 8743 to fifteen or more tiers available (15). The meaning of value 0xF is dependant on the value of the
 8744 optional the *Extended Number of Price Tiers* field. Absence of this field, or a value of 0x00 in
 8745 this field, indicates that maximum number of tiers available is fifteen. Where the *Extended*
 8746 *Number of Price Tiers* field contains a non-zero value, the maximum number of tiers available is
 8747 determined by the sum of the values of the *Number of Price Tiers* sub-field and the *Extended*
 8748 *Number of Price Tiers* field.

8749 The Register Tier values correlate which *CurrentTierNSummationDelivered* attribute, found in
 8750 sub-clause D.3.2.2.2, is accumulating usage information Register Tier enumerated values are
 8751 listed in Table D-98. The meaning of value 0xF is dependant on the value of the optional
 8752 *Extended Register Tier* field. Absence of this field, or a value of 0x00 in this field, indicates that
 8753 usage information is being accumulated in the *CurrentTier15SummationDelivered* attribute.
 8754 Where the *Extended Register Tier* field contains a non-zero value, the
 8755 *CurrentTierNSummationDelivered* attribute currently accumulating usage information by the
 8756 sum of the values of the *Register Tier* sub-field and the *Extended Register Tier* field.

8757 Both attributes can be used to calculate and display usage and subsequent costs.

8758 **Note:** Values for Register Tier Sub-field Enumerations listed above 0x6 in this revision of this
 8759 specification are provisional and not certifiable. This number of fields may change before
 8760 reaching certifiable status in a future revision of this specification.

8761

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

8762

Start Time (mandatory): A UTCTime field to denote the time at which the price signal becomes valid. A Start Time of 0x00000000 is a special time denoting “now.”

If the device would send a price with a Start Time of now, adjust the Duration In Minutes field to correspond to the remainder of the price.

Duration In Minutes (mandatory): An unsigned 16-bit field used to denote the amount of time in minutes after the Start Time during which the price signal is valid. Maximum value means “until changed”. If Block Charging only is in use (see sub-clause D.4.4.3 for further details), the Duration in Minutes field of the Publish Price command shall be set to 0xFFFF indicating the price is valid “until changed”.

Price (mandatory): An unsigned 32-bit field containing the price of the commodity measured in base unit of Currency per Unit of Measure with the decimal point located as indicated by the Price Trailing Digit field when the commodity is delivered to the premises.

Price Ratio (optional): An unsigned 8-bit field that gives the ratio of the price denoted in the Price field to the “normal” price chosen by the commodity provider. This field is thought to be useful in situations where client devices may simply be interested in pricing levels or

ratios. The value in this field should be scaled by a factor of 0.1, giving a range of ratios from 0.1 to 25.4. A value of 0xFF indicates the field is not used and 0x00 is an invalid value.

Generation Price (optional): An unsigned 32-bit field containing the price of the commodity measured in base unit of Currency per Unit of Measure with the decimal point located as indicated by the Price Trailing Digit field when the commodity is received from the premises. An example use of this field is in energy markets where the price of electricity from the grid is different than the price of electricity placed on the grid. A value of 0xFFFFFFFF indicates the field is not used.

Generation Price Ratio (optional): An unsigned 8-bit field that gives the ratio of the price denoted in the Generation Price field to the “normal” price chosen by the commodity provider. This field is thought to be useful in situations where client devices may simply be interested in pricing levels or ratios. The value in this field should be scaled by a factor of 0.1, giving a range of ratios from 0.1 to 25.4 A value of 0xFF indicates the field is not used and 0x00 is an invalid value.

Alternate Cost Delivered (optional): An unsigned 32-bit Integer field that provides a mechanism to describe an alternative measure of the cost of the energy consumed. An example of an Alternate Cost might be the emissions of CO₂ for each kWh of electricity consumed providing a measure of the environmental cost. Another example is the emissions of CO₂ for each cubic meter of gas consumed (for gas metering). A different value for each price tier may be provided which can be used to reflect the different mix of generation that is associated with different TOU rates. A value of 0xFFFFFFFF indicates the field is not used.

Alternate Cost Unit (optional): An 8-bit enumeration identifying the unit (as specified in Table D-99) for the Alternate Cost Delivered field. A value of 0xFF indicates the field is not used.

Table D-99—Alternate Cost Unit Enumerations

Values	Description
0x00	Reserved for future use
0x01	Kg of CO ₂ per unit of measure
0x02 to 0xFF	Reserved for future use

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.

8811 For combined Block/TOU charging, where multiple sets of Block Thresholds are being utilized,
 8812 the field shall indicate the number of block thresholds available in the current price tier.

8813 **Price Control (optional):** Identifies additional control options for the price event. A value of
 8814 0x00 indicates field not used. Note that for ZigBee SE 1.1 and later devices, the *Price*
 8815 *Acknowledgement* command is mandatory, but for SE 1.0 devices, it was optional, so the
 8816 sender of the *Publish Price* command should not rely on receiving a *Price Acknowledgment*
 8817 command even if the Price Acknowledgement bit in the Price Control Field is set.

8818 If Bit 1 is set, this indicates that the total number of tiers exceeds the 15 specified in the
 8819 command; this shall indicate to a client complying with this specification that it should read the
 8820 total number of tiers using the *GetTariffInformation* command.

8821 The BitMap for this field is described in Table D-100.

8822

Table D-100– Price Control Field BitMap

Bit	Description
0	0=Price Acknowledgement not required, 1=Price Acknowledgement required
1	0=Total Tiers DOES NOT exceed 15 1= Total Tiers exceeds the 15 specified in the command
2 to 7	Reserved

8823
 8824 **Number of Generation Tiers (optional):** Specifies the total number of generation tiers
 8825 applicable in the current tariff, valid values are 0-48.

8826 **Generation Tier (optional):** An 8-bit enumerated value specifying the current generation tier.
 8827 See Table D-101.

8828

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

8829	0x0A	Usage accumulating in <i>CurrentTier10SummationReceived</i> attribute
8830	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
	0x27	Usage accumulating in <i>CurrentTier39SummationReceived</i> attribute
	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

Extended Number of Price Tiers (optional): Where the maximum number of price tiers available exceeds the value of 15 supported by the *Number of Price Tiers* sub-field, this enumerated field is used in conjunction with the *Number of Price Tiers* sub-field to indicate the maximum number of price tiers available. Valid values for the *Extended Number of Price Tiers* field are from 1 to 33, indicating a maximum number of tiers available from 16 to 48 respectively. A value of zero indicates that the maximum number of price tiers available is indicated by the *Number of Price Tiers* sub-field alone.

Extended Price Tier (optional): Where the current Price Tier exceeds the value of 15 supported by the *Price Tier* sub-field, this enumerated field is used in conjunction with the *Price Tier* sub-field to indicate the current Price Tier. Valid values for the *Extended Price Tier* field are from 1 to 33, indicating a current Price Tier of 16 to 48 respectively as shown in Table D-102. A value of zero indicates that the current status of the Price Tier is indicated by the *Price Tier* sub-field alone.

Table D-102— Extended Price Tier Field Enumerations

Enumerated Value	Price Tier
0x00	Refer to <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>

8845
8846 **Extended Register Tier (mandatory):** Where the current Register Tier exceeds the value of 15
8847 supported by the *Register Tier* sub-field, this enumerated field is used in conjunction with the
8848 *Register Tier* sub-field to indicate which *CurrentTierNT summationDelivered* attribute, found in
8849 sub-clause D.3.2.2.2, is accumulating usage information. Valid values for the *Extended Register*
8850 *Tier* field are from 1 to 33, indicating a current Register Tier of 16 to 48 respectively as shown in
8851 Table D-103. A value of zero indicates that the current status of the Register Tier is indicated by
8852 the *Register Tier* sub-field alone.

8853 **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

8854

8855 D.4.2.4.1.2 Effect on Receipt8856 On receipt of this command, the device is informed of a price event for the specific
8857 provider, commodity, and currency indicated.8858 Should the device choose to change behavior based on the price event, the change of behavior
8859 should occur after a random delay between 0 and 5 minutes, to avoid potential spikes that could
8860 occur as a result of coordinated behavior changes. Likewise, should a device choose to
8861 change behavior based on the expiration of the price event, the change in behavior should occur
8862 after a random delay between 0 and 5 minutes.

8863

8864 D.4.2.4.2 **Publish Block Period Command**8865 The *Publish Block Period* command is generated in response to receiving a *Get Block*
8866 *Period(s)* command (see sub-clause D.4.2.3.5) or when an update to the block tariff schedule is
8867 available from the commodity provider. When the *Get Block Period(s)* command is received
8868 over the ZigBee Smart Energy network, the *Publish Block Period* command(s) should be sent
8869 unicast to the requestor. In the case of an update to the block tariff schedule from the

8870 commodity provider, the *Publish Block Period* command should be unicast to all individually
 8871 registered devices implementing the Price Cluster on the ZigBee Smart Energy network.

8872 Devices capable of receiving this command must be capable of storing and supporting two
 8873 block periods, the current active block and the next block. By supporting two block periods,
 8874 receiving devices will allow the *Publish Block Period* command generator to publish the
 8875 next block information during the current block period.

8876 **Note:** The *Publish Block Period* command in this revision of this specification is provisional and
 8877 not certifiable. This feature may change before reaching certifiable status in a future revision of
 8878 this specification.

8879 D.4.2.4.2.1 [Payload Format](#)

Octets	4	4	4	3	1	1	1	1
Data Type	Unsigned 32 bit Integer	Unsigned 32 bit Integer	UTCTime	Unsigned 24 bit Integer	8 bit BitMap	8 bit BitMap	8 bit BitMap	8 bit Enumeration
Field Name	Provider ID (M)	Issuer Event ID (M)	Block Period Start Time (M)	Block Period Duration (M)	Block Period Control (M)	Block Period Duration Type (M)	Tariff Type (M)	Tariff Resolution Period (M)

8880 **Figure D-74—Publish Block Period Command Payload**

8881 **Note:** M = Mandatory field, O = Optional field. All fields shall be present in the payload.
 8882 Optional fields will be marked with specific values to indicate they are not being used.

8883 **Provider ID (mandatory):** An unsigned 32-bit field containing a unique identifier for
 8884 the commodity provider. This field allows differentiation in deregulated markets where multiple
 8885 commodity providers may be available.

8886 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. When
 8887 new block period information is provided that replaces older information for the same
 8888 period, this field allows devices to determine which information is newer. It is expected that
 8889 the value contained in this field is a unique number managed by upstream servers or a
 8890 UTC based time stamp (UTCTime data type) identifying when the *Publish Block Period*
 8891 command was issued. Thus, newer block period information will have a value in the Issuer Event
 8892 ID field that is larger than older block information.

8893 **Block Period Start Time (mandatory):** A UTCTime field to denote the time at which the
 8894 block tariff period starts. A start time of 0x00000000 is a special time denoting “now”. If the
 8895 device would send an event with a Start Time of now, adjust the Duration In Minutes field
 8896 to correspond to the remainder of the event. A start date/time of 0xFFFFFFFF shall cause an
 8897 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).

Where the *Duration Timebase* is set to a value other than *Minutes*, the *Duration Control* sub-field provides further clarification; where *Duration Control* is set to *Start of Timebase*, the *Block Period Start Time* shall be set to 00:00:00 on the applicable date, and where *Duration Control* is set to *End of Timebase*, the *Block Period Start Time* shall be set to 23:59:59 on the applicable date.

Block Period Duration (mandatory): An unsigned 24-bit field to denote the block tariff period. The duration units are defined by the *Block Period Duration Type* field. Maximum value (0xFFFFFFF) means 'until changed'.

Block Period Control (mandatory): Identifies additional control options for the block period event. A value of 0x00 indicates field not used.

The BitMap for this field is described in Table D-104.

Table D-104—Block Period Control Field BitMap

Bit	Description
0	1=Price Acknowledgement required, 0=Price Acknowledgement not required
1	1=Repeating Block, 0=Non Repeating Block
2-7	Reserved

Price Acknowledgement: Indicates whether a *Price Acknowledgment* command shall be returned on receipt of this *Publish Block Period* command.

Repeating Block: Indicates whether a block period repeats on expiry. Note that the interaction between Block and Billing periods is out of scope of this specification.

Block Period Duration Type (mandatory): An 8-bit bitmap where the least significant nibble is an enumerated sub-field indicating the time base used for the duration, and the most significant nibble is an enumerated sub-field providing duration control.

Enumerated values for the Duration Timebase are shown in Table D-105:

Table D-105—Block Period DurationTimebase Enumeration

Value	Description
0x0	Minutes (default)

0x1	Days
0x2	Weeks
0x3	Months
0x4-0xF	Reserved

8923

8924 Enumerated values for the Duration Control are shown in Table D-106:

8925

Table D-106– Block Period Duration Control Enumeration

Value	Description
0x0	Start of Timebase
0x1	End of Timebase
0x2	Not Specified
0x3-0xF	Reserved

8926 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.
8927
8928
8929
8930

8931

8932 **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.8936 **Tariff Resolution Period (mandatory):** An 8 bit enumeration identifying the resolution period for the block tariff. See Table D-107:
8937

8938

Table D-107– Tariff Resolution Period Enumeration

Value	Description
0x00	Not Defined
0x01	Block Period
0x02	1 Day
0x03-0xFF	Reserved

8939 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.
89408941 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
8942

8943 end of the Block Period but smoothing out the tariff application for the customer. This is
 8944 described as follows:

8945 *Daily resolution of block tariffs is a method by which customers on a block tariff are charged on*
 8946 *the basis of assigning the block thresholds on a day in proportion to the period through the block*
 8947 *period. For example, if the Block Period is 90 days and the day is number 45 in the period, then*
 8948 *the thresholds which determine the cost to date on that day will be 50% of the thresholds defined*
 8949 *for the whole Block period. This creates an averaging effect on the block tariff and prevents the*
 8950 *customer from being exposed to one or more potentially large cost changes for many days during*
 8951 *the billing period which can create customer concern, particularly in prepayment applications,*
 8952 *and replacing these with cost changes during each day which are less apparent, but create the*
 8953 *same total charges.*

8954

8955 D.4.2.4.3 **PublishConversionFactor Command**

8956 The *PublishConversionFactor* command is sent in response to a *GetConversionFactor* command
 8957 or if a new conversion factor is available.

8958 Clients shall be capable of storing at least two instances of the Conversion Factor, the currently
 8959 active one and the next one.

8960 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)

8961 **Figure D-75 – PublishConversionFactor Command Payload**

8962

8963 D.4.2.4.3.2 Payload Details

8964 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider.

8965 **Start Time (mandatory):** A UTCTime field to denote the time at which the value becomes valid.
 8966 The value remains valid until replaced by a newer one.

8967 **Conversion Factor (mandatory):** See Price Cluster Commodity attributes (see sub-clause
 8968 D.4.2.2.4.3).

8969 **Conversion Factor Trailing Digit (mandatory):** See Price Cluster Commodity attributes (see
 8970 sub-clause D.4.2.2.4.4).

8971

8972 D.4.2.4.4 **PublishCalorificValue Command**

8973 The *PublishCalorificValue* command is sent in response to a *GetCalorificValue* command or if
 8974 a new calorific value is available. Clients shall be capable of storing at least two instances of
 8975 the Calorific Value, the currently active one and the next one.

8976 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)

8977 **Figure D-76– PublishCalorificValue Command Payload**8978 D.4.2.4.4.2 Payload Details

8979 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider.

8980 **Start Time (mandatory):** A UTCTime field to denote the time at which the value becomes valid.
 8981 The value remains valid until replaced by a newer one.

8982 **Calorific Value (mandatory):** See Price Cluster Commodity attributes (see sub-clause
 8983 D.4.2.2.4.5).

8984 **Calorific Value Unit (mandatory):** See Price Cluster Commodity attributes (see sub-clause
 8985 D.4.2.2.4.6).

8986 **Calorific Value Trailing Digit (mandatory):** See Price Cluster Commodity attributes (see
 8987 sub-clause D.4.2.2.4.7).

8988

8989 D.4.2.4.5 **PublishTariffInformation Command**

8990 The *PublishTariffInformation* command is sent in response to a *GetTariffInformation* command
 8991 or if new tariff information is available (including Price Matrix and Block Thresholds).

8992 Clients should be capable of storing at least **two** instances of the Tariff Information, the currently
 8993 active and the next one. Note that there may be separate tariff information for consumption
 8994 delivered and received.

8995 Note that the payload for this command could be up to 61 bytes in length, therefore
 8996 fragmentation may be required.

8997 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)

8998

1	2	1	4	1	3	3
8-bit Enumeration	Unsigned 16-bit Integer	8-bit BitMap	Unsigned 32-bit Integer	Unsigned 8-bit Integer	Unsigned 24-bit Integer	Unsigned 24-bit Integer
Unit of Measure (M)	Currency (M)	Price Trailing Digit (M)	Standing Charge (M)	TierBlockMode (M)	Block Threshold Multiplier (M)	Block Threshold Divisor (M)

Figure D-77 – PublishTariffInformation Command Payload

9000 D.4.2.4.5.2 Payload Details

9001 **ProviderID (mandatory):** A unique identifier for the commodity supplier. The *ProviderID* in
 9002 this command will always be the one stored as the attribute (see D.10.2.2.1.1 or D.10.2.2.1.8
 9003 depending on *TariffType*) except for the case where a change of supplier is pending and the new
 9004 supplier wishes to publish its tariff information in advance.

9005 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. When
 9006 new information is provided that replaces older information for the same time period, this field
 9007 allows devices to determine which information is newer. The value contained in this field is a
 9008 unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)
 9009 identifying when the Publish command was issued. Thus, newer information will have a value in
 9010 the Issuer Event ID field that is larger than older information.

9011 **Issuer Tariff ID (mandatory):** Unique identifier generated by the commodity supplier.

9012 **Start Time (mandatory):** A UTC Time field to denote the time at which the price signal
 9013 becomes valid. A start date/time of 0x00000000 shall indicate that the command should be
 9014 executed immediately.

9015 **Tariff Type/Charging Scheme (mandatory):** An 8-bit bitmap identifying the type of tariff
 9016 published in this command. The least significant nibble represents an enumeration of the tariff
 9017 type as detailed in Table D-108 (Generation Meters shall use the ‘Received’ Tariff), the most
 9018 significant nibble represents an enumeration specifying the charging scheme as detailed in Table
 9019 D-109.

9020

Table D-108– Tariff Type Enumeration

Value	Description
0x0	Delivered Tariff
0x1	Received Tariff
0x2	Delivered and Received Tariff
0x3-0xF	Reserved

9021

9022

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.

9023

9024 **Tariff Label (mandatory):** The format and use of this field is the same as for the *TariffLabel* attribute or *ReceivedTariffLabel* attribute (depending on *TariffType*) as defined in 347D.4.2.2.7.1 and D.4.2.2.15.1 respectively.

9027 **Number of Price Tiers in Use (mandatory):** The format and use of this field is the same as for the *NumberofPriceTiersInUse* attribute or *ReceivedNumberofPriceTiersInUse* attribute (depending on *TariffType/Charging Scheme*) as defined in D.4.2.2.7.2 and D.4.2.2.15.2 respectively.

9031 **Number of Block Thresholds in Use (mandatory):** The format and use of this field is the same as for the *NumberofBlockThresholdsInUse* attribute or *ReceivedNumberofBlockThresholdsInUse* attribute (depending on *TariffType/Charging Scheme*) as defined in D.4.2.2.7.3 and D.4.2.2.15.3 respectively.

9035 **Unit of Measure (mandatory):** The format and use of this field is the same as for the *Unit of Measure* attribute as defined in D.4.2.2.7.5.

9037 **Currency (mandatory):** The format and use of this field is the same as for the *Currency* attribute as defined in D.4.2.2.7.6.

9039 **Price Trailing Digit (mandatory):** The format and use of this field is the same as for the *PriceTrailingDigit* attribute as defined in D.4.2.2.7.7.

9041 **Standing Charge (mandatory):** The format and use of this field is the same as for the *StandingCharge* attribute as defined in D.4.2.2.4.2. A value of 0xFFFFFFFF indicates the field is not used. When publishing Received tariffs (according to *TariffType*) this field should be set to 0xFFFFFFFF.

9045 **TierBlockMode (mandatory):** The format and use of this field is the same as for the
9046 *TierBlockMode* attribute or *ReceivedTierBlockMode* attribute (depending on *TariffType*) as
9047 defined in D.4.2.2.7.4  and D.4.2.2.15.4 respectively. In case of TOU or Block Charging only,
9048 this field is not used and shall be set to 0xFF. For combined Block/TOU charging, this field is
9049 mandatory and must be set to a valid value.

9050 **BlockThresholdMultiplier (mandatory):** *BlockThresholdMultiplier* provides a value to be
9051 multiplied against Threshold parameter(s). If present, this attribute must be applied to all Block
9052 Threshold values to derive values that can be compared against the
9053 *CurrentBlockPeriodConsumptionDelivered* attribute within the Metering cluster. This parameter
9054 must be used in conjunction with the *BlockThresholdDivisor* parameter(s). In case no multiplier
9055 is defined, this field shall be set to 1.

9056 **BlockThresholdDivisor (mandatory):** *BlockThresholdDivisor* provides a value to divide the
9057 result of applying the *ThresholdMultiplier* attribute to Block Threshold values to derive values
9058 that can be compared against the *CurrentBlockPeriodConsumptionDelivered* attribute within the
9059 Metering cluster. This attribute must be used in conjunction with the *BlockThresholdMultiplier*
9060 parameter(s). In case no divisor is defined, this field shall be set to 1.

9061 **D.4.2.4.5.3 Effect on Receipt**

9062 If the client is unable to store this *PublishTariffInformation* command, the device should respond
9063 using a ZCL Default Response with a status of INSUFFICIENT_SPACE.

9064

9065 **D.4.2.4.6 PublishPriceMatrix Command**

9066 The *PublishPriceMatrix* command is used to publish the Block Price Information Set (up to 15
9067 tiers x 15 blocks) and the Extended Price Information Set (up to 48 tiers). The
9068 *PublishPriceMatrix* command is sent in response to a *GetPriceMatrix* command.

9069 Clients should be capable of storing at least two instances of the Price Matrix, the currently
9070 active and the next one.

9071 There may be a separate Price Matrix for consumption delivered and received; in this case, each
9072 Price Matrix will be identified by a different *IssuerTariffId* value.

9073 The Price server shall send only the number of tiers and blocks as defined in the corresponding
9074 *PublishTariffInformation* command (NumberofPriceTiersinUse,
9075 NumberofBlockThresholdsinUse+1).

9076 The maximum application payload may not be sufficient to transfer all Price Matrix elements in
9077 one command. Therefore the ESI may send as many *PublishPriceMatrix* commands as needed.
9078 In this case the first command shall have *CommandIndex* set to 0, the second to 1 and so on; all

9079 associated commands shall use the *same value of Issuer Event ID*. Note that, in this case, it is the
 9080 client's responsibility to ensure that it receives all associated *PublishPriceMatrix* commands
 9081 before any of the payloads can be used.

9082 D.4.2.4.6.1 [Payload Format](#)

9083 The *PublishPriceMatrix* command shall be formatted as illustrated in Figure D-78:

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

9084 **Figure D-78—*PublishPriceMatrix* Command Payload**

9085 D.4.2.4.6.2 [Payload Details](#)

9086 **Provider ID (mandatory):** An unsigned 32-bit field containing a *unique identifier for*
 9087 *the commodity provider*. This field allows differentiation in deregulated markets where multiple
 9088 commodity providers may be available.

9089 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. When
 9090 new information is provided that replaces older information for the same time period, this field
 9091 allows devices to determine which information is newer. The value contained in this field is a
 9092 unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)
 9093 identifying when the Publish command was issued. Thus, newer information will have a value in
 9094 the Issuer Event ID field that is larger than older information.

9095 **Start Time (mandatory):** A UTC Time field to denote the time at which the price signal
 9096 becomes valid. A start date/time of 0x00000000 shall indicate that the command should be
 9097 executed immediately.

9098 **Issuer Tariff ID (mandatory):** Unique identifier generated by the commodity supplier. This
 9099 must match the *Issuer Tariff ID* sent in the related *PublishTariffInformation* command.

9100 **Command Index (mandatory):** The *Command Index* is used *to count the payload fragments in*
 9101 *the case that an entire payload does not fit into one message*. The *Command Index* starts at 0 and
 9102 is incremented for each fragment belonging to the same command.

9103 **Total Number of Commands (mandatory):** In the case that an entire payload does not fit into
 9104 one message, the *Total Number of Commands* field indicates the total number of sub-commands
 9105 in the message.

9106 **Sub-Payload Control (mandatory):** An 8-bit bitmap, the least significant bit of which specifies
 9107 the information type stored in the sub payload (see Table D-110). The remaining bits are
 9108 reserved.

9109 **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

9110

9111 **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)	...

9112 **Figure D-79– PriceMatrix Command Sub-Payload**

9113 **Tier/Block ID (Mandatory):** The Tier/Block ID specifies the TOU Tier or the TOU Tier and
 9114 Block that the subsequent Price field in the command applies to. If Bit 0 of the Sub-Payload
 9115 Control field is set to Zero, then the least significant nibble represents a value specifying the
 9116 block number and the most significant nibble represents the Tier that the subsequent Price field
 9117 applies to. Valid values for the Block Number sub-field are 0 to 15 reflecting block 1 (0) to block
 9118 16(15).Valid values for the Tiers sub-field are from 0 to 15 reflecting no tiers to tier fifteen.

9119 If Bit 0 of the Sub-Payload Control field is set to one, then the field is an 8-bit value specifying
 9120 the TOU Tier that the subsequent Price field applies to. Valid values are 1 to 48.

9121 **Price (Mandatory):** This field holds the price information for the Block/TOU or TOU identified
 9122 by the previous Tier/Block ID field. The price information is provided in a base unit of *Currency*
 9123 with the decimal point located as indicated by the *Trailing Digits* field of a
 9124 PublishTariffInformation command or by the attribute defined in the *Tariff Information Attribute*
 9125 Set.

9126 **NOTE:** The number of blocks in use is one greater than the number of block thresholds in use.
 9127 For TOU charging only (number of block thresholds in use = 0, number of blocks in use = 1),
 9128 the price information of block 1, tier 1 to 15 shall be used.

9129

9130 **D.4.2.4.7 PublishBlockThresholds Command**

9131 The PublishBlockThresholds command is sent in response to a GetBlockThresholds command.

9132 Clients should be capable of storing at least **two** instances of the Block Thresholds, the currently
 9133 active and the next one.

9134 There may be a separate set of Block Thresholds for consumption delivered and received; in this
 9135 case, each set of Block Thresholds will be identified by a different *IssuerTariffId* value.

9136 The price server shall send only the number of block thresholds in use
 9137 (*NumberofBlockThresholdsInUse*) as defined in the *PublishTariffInformation* command.

9138 The maximum application payload may not be sufficient to transfer all thresholds in one
 9139 command. In this case the Price server may send two consecutive *PublishBlockThreshold*
 9140 commands (*CommandIndex* set to 0 and 1 respectively); both commands shall use the same value
 9141 of *Issuer Event ID*. Note that, in this case, it is the client's responsibility to ensure that it receives
 9142 all associated *PublishBlockThreshold* commands before any of the payloads can be used.

9143 **D.4.2.4.7.1 Payload Format**

9144 The *PublishBlockThresholds* command shall be formatted as illustrated in Figure D-80:

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

9145 **Figure D-80—*PublishBlockThresholds* Command Payload**

9146 **D.4.2.4.7.2 Payload Details**

9147 **Provider ID (mandatory):** An unsigned 32-bit field containing a unique identifier for
 9148 the commodity provider. This field allows differentiation in deregulated markets where multiple
 9149 commodity providers may be available.

9150 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. When
 9151 new information is provided that replaces older information for the same time period, this field
 9152 allows devices to determine which information is newer. The value contained in this field is a
 9153 unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)
 9154 identifying when the Publish command was issued. Thus, newer information will have a value in
 9155 the *Issuer Event ID* field that is larger than older information.

9156 **Start Time (mandatory):** A UTC Time field to denote the time at which the price signal
 9157 becomes valid. A start date/time of 0x00000000 shall indicate that the command should be
 9158 executed immediately.

9159 **Issuer Tariff ID (mandatory):** Unique identifier generated by the commodity supplier. This
 9160 must match the *Issuer Tariff ID* sent in the related *PublishTariffInformation* command.

9161 **Command Index (mandatory):** The *Command Index* is used to count the payload fragments in
 9162 the case where the entire payload does not fit into one message. The *Command Index* starts at 0
 9163 and is incremented for each fragment belonging to the same command.

9164 **Total Number of Commands (mandatory):** In the case where the entire payload does not fit
 9165 into one message, the *Total Number of Commands* field indicates the total number of sub-
 9166 commands in the message.

9167 **Sub-Payload Control (Mandatory):** The *Sub-Payload Control* bitmap specifies the usage of the
 9168 information contained within the *Block Threshold Sub-Payload* (see Table D-111).

9169 **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

9170 **D.4.2.4.7.2.1 BlockThreshold Sub-Payload**

9171 The *BlockThreshold Sub-Payload* consists of multiple sets of data which consist of a Tier ID,
 9172 Block Threshold Count and the threshold values associated with the stated Tier. The number of
 9173 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	...

9174 **Figure D-81– BlockThreshold Sub-Payload Format**

9175 **Tier/NumberOfBlockThresholds:** The *Tier/NumberOfBlockThresholds* field is an 8 bitmap.
 9176 The format of the bitmap is decided by bit0 of the sub-payload control field.

9177 If Bit0 of the *Sub-Payload Control* field is 0, then the least significant nibble represents a value
 9178 specifying the number of thresholds to follow in the command. The most significant nibble
 9179 represents the Tier that the subsequent block threshold values apply to.

9180 If Bit0 of the *Sub-Payload Control* field is 1, then the most significant nibble is unused and
 9181 should be set to 0.

9182 Valid values for the *NumberOfBlockThresholds* sub-field are 0 to 15 reflecting no block in use
 9183 (0) to block 15(15).Valid values for the Tiers sub-field are from 0 to 15 reflecting no tier to tier
 9184 fifteen.

9185 If the thresholds for a particular tier (Bit0 of the *Sub-Payload Control* field is 0) or the total
 9186 number of thresholds (Bit0 of the *Sub-Payload Control* field is 1) will not fit into a single
 9187 *PublishBlockThresholds* command, then the value of this *NumberOfBlockThresholds* sub-field
 9188 shall indicate the number of thresholds of the relevant type contained within this particular
 9189 command only.

9190 **BlockThreshold:** The Block Thresholds represent the threshold values applicable to an
 9191 individual block period and, where applicable, to a particular tier.

9192 The thresholds are established such that crossing the threshold of energy consumption for the
 9193 present block activates the next higher block, which can affect the energy rate in a positive or
 9194 negative manner. The values are absolute and always increasing. The values represent the
 9195 threshold at the end of a block. The Unit of Measure will be based on the fields defined in the
 9196 *PublishTariffInformation* command, the formatting being defined by *ThresholdDivisor* and
 9197 *ThresholdMultiplier*.

9198

9199 **D.4.2.4.8 PublishCO₂Value Command**

9200 The *PublishCO₂Value* command is sent in response to a *GetCO₂Value* command or if a new CO₂
 9201 conversion factor is available.

9202 Clients should be capable of storing at least **two** instances of the CO₂ conversion factor, the
 9203 currently active and the next one.

9204 **D.4.2.4.8.1 Payload Format**

9205 The *PublishCO₂Value* command shall be formatted as illustrated in Figure D-82:

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)

9206 **Figure D-82– PublishCO₂Value Command Payload**

9207 D.4.2.4.8.2 Payload Details

9208 **Provider ID (mandatory):** An unsigned 32-bit field containing a unique identifier for
9209 the commodity provider. This field allows differentiation in deregulated markets where multiple
9210 commodity providers may be available.

9211 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. When
9212 new information is provided that replaces older information for the same time period, this field
9213 allows devices to determine which information is newer. The value contained in this field is a
9214 unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)
9215 identifying when the Publish command was issued. Thus, newer information will have a value in
9216 the *Issuer Event ID* field that is larger than older information.

9217 **Start Time (mandatory):** A UTC Time field to denote the time at which the CO₂ value becomes
9218 valid. A start date/time of 0x00000000 shall indicate that the command should be executed
9219 immediately. A start date/time of 0xFFFFFFFF shall cause an existing *PublishCO₂Value*
9220 command with the same *Provider ID* and *Issuer Event ID* to be cancelled (note that, in markets
9221 where permanently active price information is required for billing purposes, it is recommended
9222 that a replacement/superseding *PublishCO₂Value* command is used in place of this cancellation
9223 mechanism).

9224 **Tariff Type (mandatory):** An 8-bit bitmap identifying the type of tariff published in this
9225 command. The least significant nibble represents an enumeration of the tariff type as detailed in
9226 Table D-108 (Generation Meters shall use the ‘Received’ Tariff). The most significant nibble is
9227 reserved.

9228 **CO₂ Value (mandatory):** The format and use of this field is the same as for the *CO₂* attribute or
9229 *ReceivedCO₂* attribute (depending on *TariffType*) as defined in D.4.2.2.7.9 and D.4.2.2.15.6
9230 respectively. A value of 0xFFFFFFFF indicates field not used.

9231 **CO₂ Unit (mandatory):** The format and use of this field is the same as for the *CO₂Unit* attribute
9232 or *ReceivedCO₂Unit* attribute (depending on *TariffType*) as defined in D.4.2.2.7.10 and
9233 D.4.2.2.15.7 respectively. A value of 0xFF indicates field not used.

9234 **CO₂ Trailing Digit (mandatory):** The format and use of this field is the same as for the *CO₂TrailingDigit* attribute or
9235 *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.

9237

9238 D.4.2.4.9 PublishTierLabels Command

9239 The *PublishTierLabels* command is generated in response to receiving a *GetTierLabels*
9240 command or when there is a tier label change.

9241 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

9242

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)

Figure D-83– PublishTierLabels Command Payload9244 D.4.2.4.9.2 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. 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 command was issued. Thus, newer information will have a value in the *Issuer Event ID* field that is larger than older information.

Issuer Tariff ID (mandatory): Unique identifier generated by the commodity supplier. This is used to identify the tariff that the labels apply to.

Command Index (mandatory): The *Command Index* is used to count the payload fragments in the case where the entire payload does not fit into one message. The *Command Index* starts at 0 and is incremented for each fragment belonging to the same command.

Total Number of Commands (mandatory): In the case where the entire payload does not fit into one message, the *Total Number of Commands* field indicates the total number of sub-commands in the message.

Number of Labels (mandatory): The number of *Tier ID/Tier Label* sets contained within the command.

9264 **Tier ID (mandatory):** The tier number that the associated *Tier Label* applies to.

9265 **Tier Label (mandatory):** ZCL Octet String field capable of storing a 12 character string (the
9266 first character indicates the string length, represented in hexadecimal format) encoded in the
9267 UTF-8 format.

9268

9269 D.4.2.4.10 **PublishBillingPeriod Command**

9270 The *PublishBillingPeriod* command is generated in response to receiving a *GetBillingPeriod(s)*
9271 command or when an update to the Billing schedule is available from the commodity supplier.

9272 Nested and overlapping *PublishBillingPeriod* commands are not allowed. In the case of
9273 overlapping billing periods, the period with the newer *IssuerEventID* takes priority over all
9274 nested and overlapping periods. All existing periods that overlap, even partially, should be
9275 removed. Note however that there may be separate billing schedules for consumption delivered
9276 and received.

9277 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)

9278 **Figure D-84—PublishBillingPeriod Command Payload**

9279 D.4.2.4.10.2 Payload Details

9280 **Provider ID (mandatory):** An unsigned 32-bit field containing a unique identifier for
9281 the commodity provider. This field allows differentiation in deregulated markets where multiple
9282 commodity providers may be available.

9283 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. When
9284 new information is provided that replaces older information for the same time period, this field
9285 allows devices to determine which information is newer. The value contained in this field is a
9286 unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)
9287 identifying when the Publish command was issued. Thus, newer information will have a value in
9288 the *Issuer Event ID* field that is larger than older information.

9289 **Billing Period Start Time (mandatory):** A UTC Time field to denote the time at which the
9290 billing period starts. A start time of **0x00000000** is a special time denoting “now”. A start
9291 date/time of **0xFFFFFFFF** shall cause an existing *PublishBillingPeriod* command with the same
9292 *Provider ID* and *Issuer Event ID* to be cancelled (note that, in markets where permanently active

9293 price information is required for billing purposes, it is recommended that a
 9294 replacement/superseding *PublishBillingPeriod* command is used in place of this cancellation
 9295 mechanism).

9296 **Billing Period Duration (mandatory):** An unsigned 24-bit field to denote the billing period
 9297 duration. The duration units are defined by the *Billing Period Duration Type* field.

9298 Billing periods are always repeating, i.e. after *BillingPeriodDuration* has elapsed since a
 9299 *BillingPeriodStartTime*, a new billing period will start with the same duration.

9300 **Billing Period Duration Type (mandatory):** An 8-bit bitmap where the least significant nibble
 9301 is an enumerated sub-field indicating the time base used for the duration, and the most significant
 9302 nibble is an enumerated sub-field providing duration control. Enumerated values for the Duration
 9303 Timebase are shown in Table D-105. Enumerated values for the Duration Control are shown in
 9304 Table D-106.  

9305 **Tariff Type (mandatory):** An 8-bit bitmap identifying the type of tariff published in this
 9306 command. The least significant nibble represents an enumeration of the tariff type as detailed in
 9307 Table D-108 (Generation Meters shall use the ‘Received’ Tariff). The most significant nibble is
 9308 reserved.

9309

9310 D.4.2.4.11 PublishConsolidatedBill Command

9311 The *PublishConsolidatedBill* command is used to make consolidated billing information from
 9312 previous billing periods available to other end devices. This command is issued in response to a
 9313 *GetConsolidatedBill* command or if new billing information is available.

9314 Nested and overlapping *PublishConsolidatedBill* commands are not allowed. In the case of
 9315 overlapping consolidated bills, the bill with the newer *IssuerEventID* takes priority over all
 9316 nested and overlapping bills. All existing bills that overlap, even partially, should be removed.
 9317 Note however that there may be separate consolidated bills for consumption delivered and
 9318 received.

9319 A server device shall be capable of storing **five** consolidated bill command events as a minimum.

9320 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)
------------	-----------------	---------------------	-------------------------------	-----------------------------	----------------------------------	-----------------	-----------------------	--------------	-------------------------

9321

Figure D-85– PublishConsolidatedBill Command Payload9322 D.4.2.4.11.2 Payload Details

9323 **Provider ID (mandatory):** An unsigned 32-bit field containing a unique identifier for
 9324 the commodity provider. This field allows differentiation in deregulated markets where multiple
 9325 commodity providers may be available.

9326 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. When
 9327 new information is provided that replaces older information for the same time period, this field
 9328 allows devices to determine which information is newer. The value contained in this field is a
 9329 unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)
 9330 identifying when the Publish command was issued. Thus, newer information will have a value in
 9331 the *Issuer Event ID* field that is larger than older information.

9332 **Billing Period Start Time (mandatory):** A UTC Time field containing the start time of the
 9333 related billing period. A start date/time of 0x00000000 shall indicate that the command should be
 9334 executed immediately. A start date/time of 0xFFFFFFFF shall cause an existing
 9335 *PublishConsolidatedBill* command with the same *Provider ID* and *Issuer Event ID* to be
 9336 cancelled (note that, in markets where permanently active price information is required for
 9337 billing purposes, it is recommended that a replacement/superseding *PublishConsolidatedBill*
 9338 command is used in place of this cancellation mechanism).

9339 **Billing Period Duration (mandatory):** An unsigned 24-bit field denoting the duration of the
 9340 related billing period. The duration units are defined by the *Billing Period Duration Type* field.

9341 **Billing Period Duration Type (mandatory):** An 8-bit bitmap where the least significant nibble
 9342 is an enumerated sub-field indicating the time base used for the duration, and the most significant
 9343 nibble is an enumerated sub-field providing duration control. Enumerated values for the Duration
 9344 Timebase are shown in Table D-105. Enumerated values for the Duration Control are shown in
 9345 Table D-106.

9346 **Tariff Type (mandatory):** An 8-bit bitmap identifying the type of tariff published in this
 9347 command. The least significant nibble represents an enumeration of the tariff type as detailed in
 9348 Table D-108 (Generation Meters shall use the ‘Received’ Tariff). The most significant nibble is
 9349 reserved.

9350 **Consolidated Bill (mandatory):** An unsigned 32-bit field containing the consolidated bill value
 9351 for the stated billing period. The *Consolidated Bill* field should be provided in the same currency
 9352 as used in the Price cluster.

9353 **Currency (mandatory):** An unsigned 16-bit field containing identifying information concerning
 9354 the local unit of currency used in the *Consolidated Bill* field.

9355 The value of the currency field should match the values defined by ISO 4217.

9356 **BillTrailingDigit (mandatory):** An 8-bit field used to determine where the decimal point is
 9357 located in the *Consolidated Bill* field. The most significant nibble contains the *Trailing Digit* sub-
 9358 field which indicates the number of digits to the right of the decimal point.

9359

9360 D.4.2.4.12 **PublishCPPEvent Command**

9361 The *PublishCPPEvent* command is sent from an ESI to its Price clients to notify them of a
 9362 Critical Peak Pricing (CPP) event.

9363 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	Unsigned 8-bit Integer	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)

9364 Figure D-86—*PublishCPPEvent* Command Payload

9365 D.4.2.4.12.2 **Payload Details**

9366 **Provider ID (mandatory):** An unsigned 32-bit field containing a unique identifier for
 9367 the commodity provider. This field allows differentiation in deregulated markets where multiple
 9368 commodity providers may be available.

9369 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. When
 9370 new information is provided that replaces older information for the same time period, this field
 9371 allows devices to determine which information is newer. The value contained in this field is a
 9372 unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)
 9373 identifying when the Publish command was issued. Thus, newer information will have a value in
 9374 the *Issuer Event ID* field that is larger than older information.

9375 **Start Time (mandatory):** A UTC Time field to denote the time at which the CPP event begins.
 9376 A start date/time of 0x00000000 shall indicate that the command should be executed
 9377 immediately. A start date/time of 0xFFFFFFFF shall cause an existing *PublishCPPEvent*
 9378 command with the same *Provider ID* and *Issuer Event ID* to be cancelled (note that, in markets
 9379 where permanently active price information is required for billing purposes, it is recommended
 9380 that a replacement/superseding *PublishCPPEvent* command is used in place of this cancellation
 9381 mechanism).

9382 **Duration in Minutes:** Defines the duration of the CPP event.

9383 **Tariff Type (mandatory):** An 8-bit bitmap identifying the type of tariff published in this
9384 command. The least significant nibble represents an enumeration of the tariff type as detailed in
9385 Table D-108 (Generation Meters shall use the ‘Received’ Tariff). The most significant nibble is
9386 reserved.

9387 **CPP Price Tier (mandatory):** An 8-bit integer identifying the price tier associated with this
9388 CPP event. The price(s) contained in the active price matrix for that price tier will override the
9389 normal pricing scheme. Prices ‘CPP1’ and ‘CPP2’ are reserved for this purposes (see Table
9390 D-80).

9391 **CPP Auth (mandatory):** An 8-bit enumeration identifying the status of the CPP event:

9392 **Table D-112– CPP Auth Enumeration**

Value	Description
0	Pending
1	Accepted
2	Rejected
3	Forced
4 - 255	Reserved

9393

9394 D.4.2.4.12.3 When Generated

9395 The *PublishCPPEvent* command is generated when the energy provider has requested the
9396 consumer to accept a CPP, when the consumer has accepted the CPP, or if the ESI has received a
9397 *CPPEventResponse* command. See Figure D-87.

9398

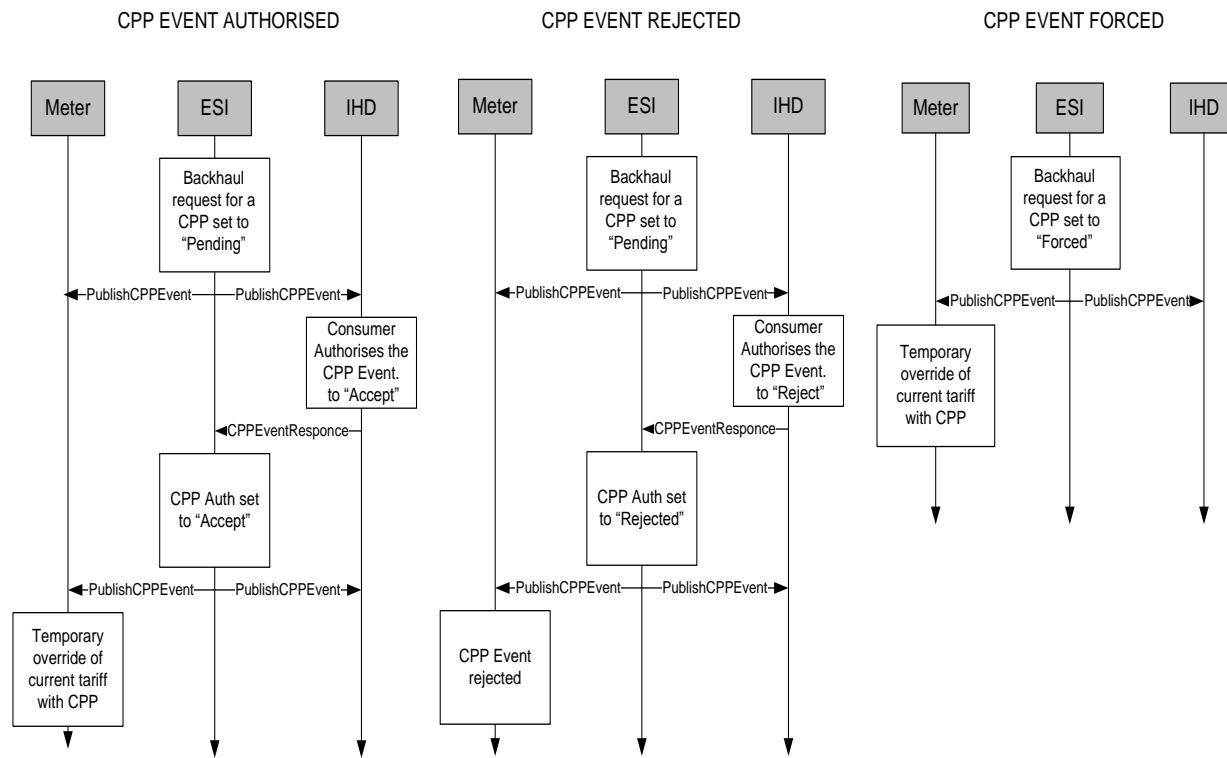


Figure D-87 – CPP Event Flow

9399
9400
94019402 D.4.2.4.12.4 Effect on Receipt9403 When the *PublishCPPEvent* command is received, the IHD or Meter shall act in one of two
9404 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 *CPPAuth* parameter set to Accepted or Rejected). It is recommended that the CPP event is ignored until a consumer either accepts or rejects the event.

9410 2. The *CPPAuth* parameter is set to "Forced", in which case the *CPPEvent* has been accepted.

9411

9412 D.4.2.4.13 PublishCreditPayment Command9413 The *PublishCreditPayment* command is used to update the credit payment information when
9414 available.

9415 Nested and overlapping *PublishCreditPayment* commands are not allowed. In the case of
 9416 overlapping credit payments, the payment with the newer *Issuer Event ID* takes priority over all
 9417 nested and overlapping payments. All existing payments that overlap, even partially, should be
 9418 removed.

9419 A server device shall be capable of storing **five** credit payments command events as a minimum.

9420 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)

9421 **Figure D-88—PublishCreditPayment Command Payload**

9422 D.4.2.4.13.2 Payload Details

9423 **Provider ID (mandatory):** An unsigned 32-bit field containing a unique identifier for
 9424 the commodity provider. This field allows differentiation in deregulated markets where multiple
 9425 commodity providers may be available.

9426 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. When
 9427 new information is provided that replaces older information for the same time period, this field
 9428 allows devices to determine which information is newer. The value contained in this field is a
 9429 unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)
 9430 identifying when the Publish command was issued. Thus, newer information will have a value in
 9431 the *Issuer Event ID* field that is larger than older information.

9432 **Credit Payment Due Date (mandatory):** A UTC Time field containing the time that the next
 9433 credit payment is due. See also section D.4.2.2.9.1. 

9434 **Credit Payment Overdue Amount (mandatory):** An unsigned 32-bit field denoting the current
 9435 amount this is overdue from the consumer. This field should be provided in the same currency as
 9436 used in the Price cluster. See also section D.4.2.2.9.3. 

9437 **Credit Payment Status (mandatory):** An 8-bit enumeration identifying the current credit
 9438 payment status. Refer to section D.4.2.2.9.2 for the format of this enumeration.


9439 **Credit Payment (mandatory):** An unsigned 32-bit field denoting the last credit payment. This
 9440 field should be provided in the same currency as used in the Price cluster. See also section
 9441 D.4.2.2.9.6. 

9442 **Credit Payment Date (mandatory):** A UTC Time field containing the time at which the last
 9443 credit payment was made. See also section D.4.2.2.9.7. 

9444 **Credit Payment Ref (mandatory):** An string of between 0-20 octets used to denote the last
 9445 credit payment reference used by the energy supplier. See also section D.4.2.2.9.8. 

9446 **D.4.2.4.14 PublishCurrencyConversion Command**

9447 The *PublishCurrencyConversion* command is sent in response to a *GetCurrencyConversion*
 9448 command or when a new currency becomes available.

9449 **D.4.2.4.14.1 Payload Format**

9450 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)

9451 **Figure D-89– *PublishCurrencyConversion* Command Payload**

9452 **D.4.2.4.14.2 Payload Details**

9453 **Provider ID (mandatory):** An unsigned 32-bit field containing a unique identifier for
 9454 the commodity provider. This field allows differentiation in deregulated markets where multiple
 9455 commodity providers may be available.

9456 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. When
 9457 new information is provided that replaces older information for the same time period, this field
 9458 allows devices to determine which information is newer. The value contained in this field is a
 9459 unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)
 9460 identifying when the Publish command was issued. Thus, newer information will have a value in
 9461 the *Issuer Event ID* field that is larger than older information.

9462 **Start Time (mandatory):** A UTC Time field to denote the time at which the new currency
 9463 becomes valid. A start date/time of 0x00000000 shall indicate that the command should be
 9464 executed immediately. A start date/time of 0xFFFFFFFF shall cause an existing but pending
 9465 *PublishCurrencyConversion* command with the same *Provider ID* and *Issuer Event ID* to be
 9466 cancelled.

9467 **Old Currency (mandatory):** An unsigned 16-bit field containing identifying information
 9468 concerning the old local unit of currency used in the Price cluster. The value of the *Old Currency*
 9469 field should match the values defined by ISO 4217.

9470 **New Currency (mandatory):** An unsigned 16-bit field containing identifying information
 9471 concerning the new local unit of currency used in the Price cluster. The value of the *New*
 9472 *Currency* field should match the values defined by ISO 4217.

9473 **Conversion Factor (mandatory):** The format and use of this field is the same as for the
 9474 *ConversionFactor* attribute as defined in D.4.2.2.4.3.

9475 **Conversion Factor Trailing Digit (mandatory):** The format and use of this field is the same as
 9476 for the *ConversionFactorTrailingDigit* attribute as defined in D.4.2.2.4.4.

9477 **Currency Change Control Flags (mandatory):** A 32-bit mask that denotes the functions that
 9478 are required to be carried out on processing of this command. See Table D-113 below:

9479 **Table D-113– 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

9480

9481 **D.4.2.4.15 CancelTariff Command**

9482 The *CancelTariff* command indicates that all data associated with a particular tariff instance
 9483 should be discarded.

9484 In markets where permanently active price information is required for billing purposes, it is
 9485 recommended that replacement/superseding *PublishTariffInformation*, *PublishPriceMatrix*,
 9486 *PublishBlockThresholds* and *PublishTierLabels* commands are used in place of a *CancelTariff*
 9487 command.

9488 **D.4.2.4.15.1 Payload Format**

9489 The *CancelTariff* command shall be formatted as illustrated in Figure D-90:

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)
------------	-----------------	----------------------	-----------------

9490

Figure D-90– CancelTariff Command Payload9491 **D.4.2.4.15.2 Payload Details**

9492 **Provider ID (mandatory):** An unsigned 32-bit field containing a unique identifier for the
 9493 commodity provider. This field allows differentiation in deregulated markets where multiple
 9494 commodity providers may be available.

9495 **Issuer Tariff ID (mandatory):** Unique identifier generated by the commodity Supplier. All
 9496 parts of a tariff instance shall have the same *Issuer Tariff ID*.

9497 **Tariff Type (mandatory):** An 8-bit bitmap identifying the *type of tariff to be cancelled by this*
 9498 *command*. The least significant nibble represents an enumeration of the tariff type as detailed in
 9499 Table D-108 (Generation Meters shall use the ‘Received’ Tariff). The most significant nibble is
 9500 reserved.

9501 **D.4.2.4.15.3 Effect on Receipt**

9502 On receipt of this command, a client device shall discard all instances of
 9503 *PublishTariffInformation*, *PublishPriceMatrix*, *PublishBlockThresholds* and *PublishTierLabels*
 9504 commands associated with the stated *Provider ID*, *Tariff Type* and *Issuer Tariff ID*.

9505 **D.4.3 Client**9506 **D.4.3.1 Dependencies**

9507 Events carried using this cluster include a timestamp with the assumption that target devices
 9508 maintain a real time clock. Devices can acquire and synchronize their internal clocks via the
 9509 ZCL Time server.

9510 If a device does not support a real time clock it is assumed that the device will interpret and
 9511 utilize the “Start Now” 0x00000000 value within the Time field.

9512 **Note:** The Price Client Cluster Attributes in this revision of this specification are provisional
 9513 and not certifiable. These features may change before reaching certifiable status in a future
 9514 revision of this specification.

9515 **D.4.3.2 Attributes**9516 **Table D-114– 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>CommodityType</i>	8-bit Enumeration	0x00 to 0xFF	Read Only	-	O
0x0003 - 0x000F	Reserved					

9517

9518 D.4.3.2.1 **PriceIncreaseRandomizeMinutes Attribute**

9519 The *PriceIncreaseRandomizeMinutes* attribute represents the maximum amount of time to be
 9520 used when randomizing the response to a price increase. Note that although the granularity of
 9521 the attribute is in minutes, it is recommended the granularity of the randomization used within
 9522 a responding device be in seconds or smaller. If a device responds to a price increase it must
 9523 choose a random amount of time, in seconds or smaller, between 0 and
 9524 *PriceIncreaseRandomizeMinutes* minutes. The device must implement that random amount of
 9525 time before or after the price change. How and if a device will respond to a price increase is up
 9526 to the manufacturer. Whether to respond before or after the price increase is also up to the
 9527 manufacturer.

9528 As an example, a water heater with a *PriceIncreaseRandomizeMinutes* set to 6 could choose to
 9529 lower its set point 315 seconds (but not more than 360 seconds) before the price increases.

9530 The valid range for this attribute is 0x00 to 0x3C.

9531 If *PriceIncreaseRandomizeMinutes* or *PriceDecreaseRandomizeMinutes* attributes
 9532 are not supported by the client, then it should use the default values for the attributes as specified
 9533 in the Price Client Cluster Attribute table.

9534 D.4.3.2.2 **PriceDecreaseRandomizeMinutes Attribute**

9535 The *PriceDecreaseRandomizeMinutes* attribute represents the maximum number of minutes to
 9536 be used when randomizing the response to a price decrease. Note that although the granularity
 9537 of the attribute is in minutes, it is recommended the granularity of the randomization used within
 9538 a responding device be in seconds or smaller. If a device responds to a price decrease it must
 9539 choose a random amount of time, in seconds or smaller, between 0 and
 9540 *PriceDecreaseRandomizeMinutes* minutes and implement that random amount of time before or
 9541 after the price change. How and if a device will respond to a price decrease is up to the
 9542 manufacturer. Whether to respond before or after the price increase is also up to the
 9543 manufacturer.

9544 As an example, a dishwasher with a *PriceDecreaseRandomizeMinutes* set to 15 could choose
9545 to start its wash cycle 723 seconds (but not more than 900 seconds) after the price decreases.

9546 The valid range for this attribute is 0x00 to 0x3C.

9547 **D.4.3.2.3 CommodityType Attribute**

9548 *CommodityType* provides a label for identifying the type of pricing client present. The attribute is
9549 an enumerated value representing the commodity. The defined values are represented by the
9550 non-mirrored values (0-127) in the *MeteringDeviceType* attribute enumerations (refer to Table
9551 D-27).

9552 **D.4.3.3 Commands Received**

9553 The client receives the cluster-specific response commands detailed in sub-clause D.4.2.4.

9554 **D.4.3.4 Commands Generated**

9555 The client generates the cluster-specific commands detailed in sub-clause D.4.2.3, as required by
9556 the application.

9557

9558 **D.4.4 Application Guidelines**

9559 **D.4.4.1 Registering for Commands**

9560 Devices should use bind request to register for unsolicited *Publish Price*, *Display Message*
9561 and *Load Control Event* commands.

9562 **D.4.4.2 Attribute Reporting**

9563 Attribute reporting may be used for sending information in the Price Server Cluster
9564 Attributes table. The Price Cluster attributes can be polled periodically for updates. Polling
9565 should not occur more frequently than recommended in D.3.4.2. Use of the *Report Attribute*
9566 command without report configuration may be used for unsolicited notification of an attribute
9567 value change. Sleepy devices may have to poll.

9568 **D.4.4.3 Block Tariffs**

9569 Upon reaching the *Start Time* of a received *Publish Price* command, a device's behavior will
9570 depend on the values of the *Number of Block Thresholds* and *Number of Price Tiers* fields. A
9571 client device needing to determine if it should use Block Pricing shall send a *Get Current Price*
9572 command to the Price server and check the *Number of Block Thresholds* in the Publish Price
9573 response. Any value between 1 and 15 indicates that Block Pricing shall be used.

9574 The prices for a commodity being delivered to the premises shall be taken from the Block Pricing
9575 Information Attribute Set whenever Block Pricing is active.

9576 D.4.4.3.1 **TOU Charging Only**

9577 Indicated by the Number of Block Thresholds field being set to zero. Charging shall be
9578 according to the price fields within the *Publish Price* command itself.

9579 D.4.4.3.2 **Block Charging only**

9580 Indicated by the *Number of Price Tiers* fields being set to zero while the *Number of Block*
9581 *Thresholds* is between 0x01 and 0x0F.

9582 A server shall not update the Block Threshold and Block Price attribute sets of an active Block
9583 Period. Updates to these attribute sets can only be done by creating a new Block Period. The
9584 server may create a new active Block Period by updating either *Block Period Start Time*
9585 (attribute *StartOfBlockPeriod*) alone or *Block Period Duration in Minutes* (attribute
9586 *BlockPeriodDuration*) followed by *Block Period Start Time* (attribute *StartOfBlockPeriod*) along
9587 with updating other attributes as desired.

9588 When a server transmits a Publish Price command it shall additionally fill fields necessary to
9589 support backwards compatibility with clients that may not support Block Charging. The *Price*
9590 field shall be set according to the Block Price Information Attribute Set. The *Duration in Minutes*
9591 field shall be set to 0xFFFF indicating the price is valid “until changed”.

9592 A server shall additionally transmit a Publish Price command to clients under the following
9593 conditions:

- 9594 1. At the start of a Block Period
- 9595 2. When it is notified that a Block Threshold has been crossed
- 9596 3. When *Block Period Start Time* or *Block Period Duration in Minutes* have changed to
9597 indicate a new active block period

9598 A client may cache attributes from the Block Threshold, Block Period, Block Price, and Billing
9599 Period attribute sets. Cached attributes are valid only during the active Block Period when
9600 received. Upon reaching *Block Period Start Time* or detecting a new active Block Period, the
9601 client should retrieve updated values for cached attributes.

9602 A client shall check for a new active Block Period on receipt of an asynchronous Publish Price
9603 command (i.e. not required on a Publish Price command in response to Get Current Price) by
9604 checking *Block Period Start Time* and *Block Period Duration in Minutes* for update.
9605 Additionally, it shall infrequently (e.g. once an hour) query the *StartOfBlockPeriod* and
9606 *BlockPeriodDuration* attributes to verify that the Block Period has not ended early.

9607 D.4.4.3.3 **Block/TOU Combination Charging**

9608 *Note: The following application guidelines that pertain to Block/TOU Combination Charging*
9609 *in this revision of this specification are provisional and not certifiable. This text may change*
9610 *before reaching certifiable status in a future revision of this specification.*

9611 The *Number of Block Thresholds* and *Number of Price Tiers* fields will both be set to non-zero
9612 values, indicating the number of blocks and number of tiers respectively being used. The
9613 start of a Block period shall be indicated by the value of the *Block Period Start Time* field within
9614 a *Publish Block Period* command. Upon reaching the *Block Period Start Time*, the attributes for
9615 the required number of Block Thresholds, together with the Block Prices for all required blocks
9616 for the selected tier should be fetched from the server. The *Block Period Duration in Minutes*
9617 field shall indicate the length of the block period.

9618 A *Publish Price* command will be received for the start of each new TOU period during a block
9619 period. At this point the attributes for the Block Prices for all required blocks for the newly
9620 activated tier should be fetched from the server.

9621 Devices shall cater for both ‘blocks in tiers’ and ‘tiers in blocks’ models. In either case, the
9622 relevant prices will be defined in the Block Pricing Information Attribute Set. The ‘tiers in
9623 blocks’ model will always implement a single set of block thresholds, whereas the ‘blocks in
9624 tiers’ model may implement different thresholds for each tier.

9625 **D.4.4.3.4 Application Guidelines for Block Pricing under specific events**

9626 **HAN device not communicating with meter for extended period of time:**

9627 In this situation, when the HAN device reconnects with the meter, it will need to read the Block
9628 Information Set to calculate the correct cost for the given period. This is done by applying
9629 the prices for each block/tier combination to the consumption information for each
9630 block/tier combination. If a block period has passed while the HAN device was not
9631 communicating with the meter, then the prior period consumption information will not be
9632 known and the prior period cost cannot be calculated by the HAN device.

9633 **Meter installation or swap-out:**

9634 The new meter will need to be configured with the appropriate block thresholds, pricing, and
9635 block duration by the utility. If this does not occur precisely at the start of that customer's
9636 billing period, the utility will need to (a) pro-rate these amounts over the remaining billing
9637 period duration and (b) decide how to handle the initial portion of the period. Any information
9638 from the initial part of the billing period will be lost when the new meter is installed. As such,
9639 HAN devices may not display accurate information for this billing period and utilities should
9640 advise customers of this situation. As a typical meter lifetime is expected to be in the range of
9641 10 to 20 years, this event is expected to be rare.

9642 **D.4.4.4 Handling of Enhanced Tariffs**

9643 In ‘Traditional’ Smart Energy networks, the back-haul connection and Price server are
9644 incorporated into the meter. Fiscal accounting is out of scope of the ZigBee network. Indicative
9645 pricing information, determined by the utility or by an ESI using information supplied by the

9646 utility, is communicated from the Price server to other Smart Energy devices, using a *Publish*
9647 *Price* command, whenever the price changes.

9648 In Smart Energy networks where the meter is detached from the back-haul connection, the meter
9649 often being battery-powered and therefore unable to communicate for the majority of the time,
9650 and specifically where a Prepayment meter is required to perform independent accounting
9651 functionality, there is a need for the meter to have local access to current price and price
9652 scheduling information at all times. The optional ‘Enhanced’ tariff mechanism described in this
9653 section provides functionality to satisfy this requirement.

9654 An enhanced tariff consists of a number of commands. Depending on the mode of operation, an
9655 associated TOU calendar may also be required. *PublishBlockThresholds* and *PublishPriceMatrix*
9656 commands always include the number of block thresholds in use and number of blocks / tiers in
9657 use respectively. It is the responsibility of a client to fetch all parts belonging to a tariff after it
9658 has received an unsolicited *PublishTariffInformation* command. A client shall ensure that it
9659 successfully receives all commands associated with a tariff before any of the data for that tariff
9660 can be used. It is recommended that a client checks that the data received across all commands is
9661 valid.

9662 Whenever a new tariff is made available to a Price Server, it shall send an unsolicited
9663 *PublishTariffInformation* command to its bound clients (BOMDs shall be notified via
9664 notification flags). Other parts of the tariff (*PriceMatrix* and *BlockThresholds*) are not sent
9665 unsolicited; the clients shall send corresponding *GetPriceMatrix* and *GetBlockThresholds*
9666 commands, as applicable, to fetch the required information from the server.

9667 The Price Cluster supports different charging modes:

- 9668 • TOU charging
- 9669 • Block charging
- 9670 • TOU/Block combination charging

9671 D.4.4.4.1 **Block Charging**

9672 In case of Block charging, the following information needs to be transferred from server to
9673 client:

- 9674 1 PublishTariffInformation
- 9675 2 PublishPriceMatrix (noTierBlock1 .. noTierBlockN)
- 9676 3 PublishBlockThresholds
- 9677 4 PublishBlockPeriod

9678 In addition, a Gas-ESI may send a *PublishConversionFactor* or *PublishCalorificValue* command
9679 along with a tariff update, but does not necessarily need to.

9680 D.4.4.4.2 **TOU Charging**

9681 In case of TOU charging, the following price information needs to be transferred from server to
9682 client:

9683 1 PublishTariffInformation

9684 2 PublishPriceMatrix (Tier1Block1 .. TierNBlock1)

9685 3 PublishCalendar (see D.9 for further details)

9686 In addition, a Gas-ESI may send a *PublishConversionFactor* or *PublishCalorificValue* command
9687 along with a tariff update, but does not necessarily need to.

9688 *Note: the TOU Calendar and the Tariff are linked by the start time and not by any IDs.*

9689 D.4.4.4.3 **TOU/Block Charging**

9690 In case of TOU/Block charging, the following price information needs to be transferred from
9691 server to client:

9692 1 PublishTariffInformation

9693 2 PublishPriceMatrix (Tier1Block1 .. TierNBlockM)

9694 3 PublishBlockThresholds

9695 4 PublishBlockPeriod

9696 5 PublishCalendar (see D.9 for further details)

9697 In addition, a Gas-ESI may send a *PublishConversionFactor* or *PublishCalorificValue* command
9698 along with a tariff update, but does not necessarily need to.

9699 D.4.4.4.4 **Critical Peak Pricing**

9700 The following additional guidelines hold for the usage of CPP events:

9701 • The price tiers used for CPP events (via the *PublishCPPEvent* command) are treated in the
9702 price matrix just like the ones used in the TOU Calendar. In fact, nothing prevents a tariff
9703 scheme where the same price is employed at regular times through the TOU calendar and ad-
9704 hoc via CPP events. Two prices are reserved in the price matrix for CPP events, ‘CPP1’ and
9705 ‘CPP2’

9706 • ESIs conforming to these specifications need to send out a *Publish Price* command along
9707 with the *PublishCPPEvent* command, for Smart Energy devices that do not support the
9708 latter.

9709 D.4.4.4.5 **Generation Charging**

9710 All Generation meters shall use the ‘Received’ sections of the Price cluster to publish the tariff
9711 information, and the Received section of the Metering Cluster.

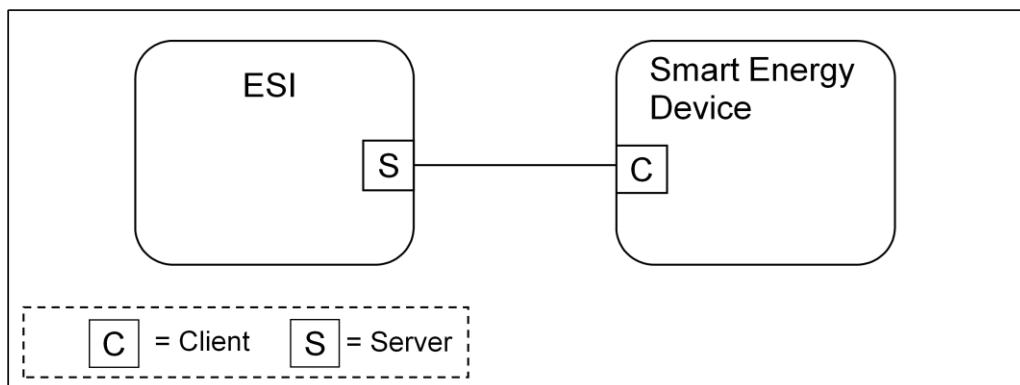
9712

9713

9714

9715 **D.5 Messaging Cluster**9716 **D.5.1 Overview**

9717 This cluster provides an interface for passing text messages between ZigBee devices.
9718 Messages are expected to be delivered via the ESI and then unicast to all individually registered
9719 devices implementing the Messaging Cluster on the ZigBee network, or just made available to
9720 all devices for later pickup. Nested and overlapping messages are not allowed. The current active
9721 message will be replaced if a new message is received by the ESI.



9722

9723 **Figure D-91– Messaging Cluster Client/Server Example**

9724 Please note the ESI is defined as the Server due to its role in acting as the proxy for upstream
9725 message management systems and subsequent data stores.

9726 **D.5.2 Server**9727 **D.5.2.1 Dependencies**

9728 Support for ZCL Data Types.

9729 No dependencies exist for other Smart Energy Clusters.

9730 **D.5.2.2 Attributes**

9731 None.

9732 **D.5.2.3 Commands Generated**

9733 The command IDs generated by the Messaging server cluster are listed in Table D-115.

9734 **Table D-115– 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	

9735

9736 D.5.2.3.1 **Display Message Command**9737 D.5.2.3.1.1 Payload Format9738 The *Display Message* command payload shall be formatted as illustrated in Figure D-92.

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)

9739

Figure D-92– Display Message Command Payload9740 D.5.2.3.1.1.1 *Payload Details*9741 **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.9745 **MessageControl (mandatory):** An 8-bit BitMap field indicating control information related to the message. Bit encoding of this field is outlined in Table D-116:

9747

Table D-116– 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.

9748

9749

If the Inter-PAN transmission mechanism outlined in Annex B is not supported on a particular device, Bits 0 to 1 can be ignored. 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.

9750

9751

9752

9753

9754

9755

9756

9757

9758

9759

9760

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.

9761

9762

9763

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.

9764

9765

9766

9767

Start Time (mandatory): A UTCTime field to denote the time at which the message becomes valid. A Start Time of 0x00000000 is a special time denoting "now." If the device would send an event with a Start Time of now, adjust the Duration In Minutes field to correspond to the remainder of the event.

9768

9769

9770

Duration In Minutes (mandatory): An unsigned 16-bit field is used to denote the amount of time in minutes after the Start Time during which the message is displayed. A Maximum value of 0xFFFF means "until changed".

9771 **Message (mandatory):** A ZCL String containing the message to be delivered. The String shall be
 9772 encoded in the UTF-8 format. Devices will have the ability to choose the methods for managing
 9773 messages that are larger than can be displayed (truncation, scrolling, etc.).

9774 For supporting larger messages sent over the network, both devices must agree upon a common
 9775 Fragmentation ASDU Maximum Incoming Transfer Size. Please refer to sub-clause 5.3.8 for
 9776 further details on Fragmentation settings.

9777 Any message that needs truncation shall truncate on a UTF-8 character boundary. The SE
 9778 secure payload is 59 bytes for the Message field in a non- fragmented, non-source routed
 9779 Display Message packet (11 bytes for other Display Message fields). Devices using
 9780 fragmentation can send a message larger than this. Reserving bytes for source route will reduce
 9781 this.

9782 **ExtendedMessageControl (optional):** An 8-bit BitMap field indicating additional control and
 9783 status information for a given message. Bit encoding of this field is shown in Table D-117:

9784

Table D-117– 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		

9785

9786 The *Message Confirmation Status* bit allows the confirmation state of a message to be
 9787 communicated in the event that there are multiple IHD's (or other Messaging cluster client
 9788 devices) on a network.

9789 The server shall initially transmit a message requiring a confirmation with the *Message*
 9790 *Confirmation Status* bit reset (0) to indicate the message had not yet been confirmed (the
 9791 *Message Confirmation* bit of the *MessageControl* field will be set to indicate a confirmation is
 9792 required).

9793 When the message is confirmed on one of the multiple IHDs in the premises, a *Message*
 9794 *Confirmation* command will be returned to the server. At this point, the server shall re-transmit
 9795 the original message, but with the *Message Confirmation Status* bit now set (1) to indicate that
 9796 the message has been confirmed. This will indicate to other clients that the message no longer
 9797 requires a confirmation.

9798

9799 D.5.2.3.2 Cancel Message Command

9800 The *Cancel Message* command described in Figure D-93 provides the ability to cancel the
 9801 sending or acceptance of previously sent messages. When this message is received the recipient
 9802 device has the option of clearing any display or user interfaces it supports, or has the option of
 9803 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)

9804 **Figure D-93– Cancel Message Command Payload**9805 D.5.2.3.2.1 Payload Details

9806 **Message ID (mandatory):** A unique unsigned 32-bit number identifier for the message being
 9807 cancelled. It's expected the value contained in this field is a unique number managed by
 9808 upstream systems or a UTC based time stamp (UTCTime data type) identifying when the
 9809 message was originally issued.

9810 **MessageControl (mandatory):** This field is deprecated and should be set to 0x00.

9811

9812 D.5.2.3.3 Display Protected Message Command

9813 The *Display Protected Message* command is for use with messages that are protected by a
 9814 password or PIN.

9815 D.5.2.3.3.1 Payload Format

9816 The payload for this command shall be the same as that for a conventional *Display Message*
 9817 command. See D.5.2.3.1.1 for payload details.

9818

9819 D.5.2.3.4 Cancel All Messages Command

9820 The *Cancel All Messages* command indicates to a client device that it should cancel all display
 9821 messages currently held by it.

9822 D.5.2.3.4.1 Payload Format

Octets	4
---------------	----------

Data Type	UTC Time
Field Name	Implementation Date/Time (M)

9823

Figure D-94—Cancel All Messages Command Payload9824 D.5.2.3.4.2 Payload Details9825 **Implementation Date/Time (mandatory):** A UTC Time field to indicate the date/time at which
9826 all existing display messages should be cleared.

9827

9828 **D.5.3 Client**9829 **D.5.3.1 Dependencies**

9830 Support for ZCL Data Types.

9831 No dependencies exist for other Smart Energy Clusters.

9832 **D.5.3.2 Attributes**

9833 None.

9834 **D.5.3.3 Commands Generated**

9835 The command IDs generated by the Messaging cluster are listed in Table D-118.

Table D-118—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	

9837

9838

9839 D.5.3.3.1 Get Last Message Command

9840 This command has no payload.

9841 D.5.3.3.1.1 Effect on Receipt

9842 On receipt of this command, the device shall send a *Display Message* or *Display Protected*
 9843 *Message* command as appropriate (refer to sub-clauses D.5.2.3.1 and D.5.2.3.3). A ZCL Default
 9844 Response with status NOT_FOUND shall be returned if no message is available.

9845

9846 D.5.3.3.2 Message Confirmation Command

9847 The *Message Confirmation* command described in Figure D-95 provides an indication that a
 9848 Utility Customer has acknowledged and/or accepted the contents of a previously sent message.
 9849 Enhanced Message Confirmation commands shall contain an answer of ‘NO’, ‘YES’ and/or a
 9850 message confirmation string.

9851 If the optional *Message Confirmation Response* is required, the *Message Confirmation Control*
 9852 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)

9853 **Figure D-95—Message Confirmation Command Payload**9854 D.5.3.3.2.1 Payload Details

9855 **Message ID (mandatory):** A unique unsigned 32-bit number identifier for the message being
 9856 confirmed.

9857 **Confirmation Time (mandatory):** UTCTime of user confirmation of message.

9858 **Message Confirmation Control (optional):** An 8-bit BitMap field indicating the simple
 9859 confirmation that is contained within the response. Bit encoding of this field is outlined in Table
 9860 D-119; if this optional field is not available, a default value of 0x00 shall be used.

9861 **Table D-119—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		

9862

Message Confirmation Response (optional): A ZCL Octet String containing the message to be returned. The first Octet indicates length. The string shall be encoded in the UTF-8 format. If this optional field is not available, a default value of 0x00 shall be used.

9866

9867 D.5.3.3.3 GetMessageCancellation Command

9868 This command initiates the return of the first (and maybe only) *Cancel All Messages* command
9869 held on the associated server, and which has an implementation time equal to or later than the
9870 value indicated in the payload.

9871 D.5.3.3.3.1 Payload Format

Octets	4
Data Type	UTC Time
Field Name	Earliest Implementation Time (M)

Figure D-96– GetMessageCancellation Command Payload

9873 D.5.3.3.2 Payload Details

9874 **Earliest Implementation Time (mandatory):** UTC Timestamp indicating the earliest
9875 implementation time of a *Cancel All Messages* command to be returned.

9876 D.5.3.3.3.3 When Generated

9877 This command is generated when the client device wishes to fetch any pending *Cancel All*
9878 *Messages* command from the server (see D.5.2.3.4 for further details). In the case of a BOMD,
9879 this may be as a result of the associated Notification flag.

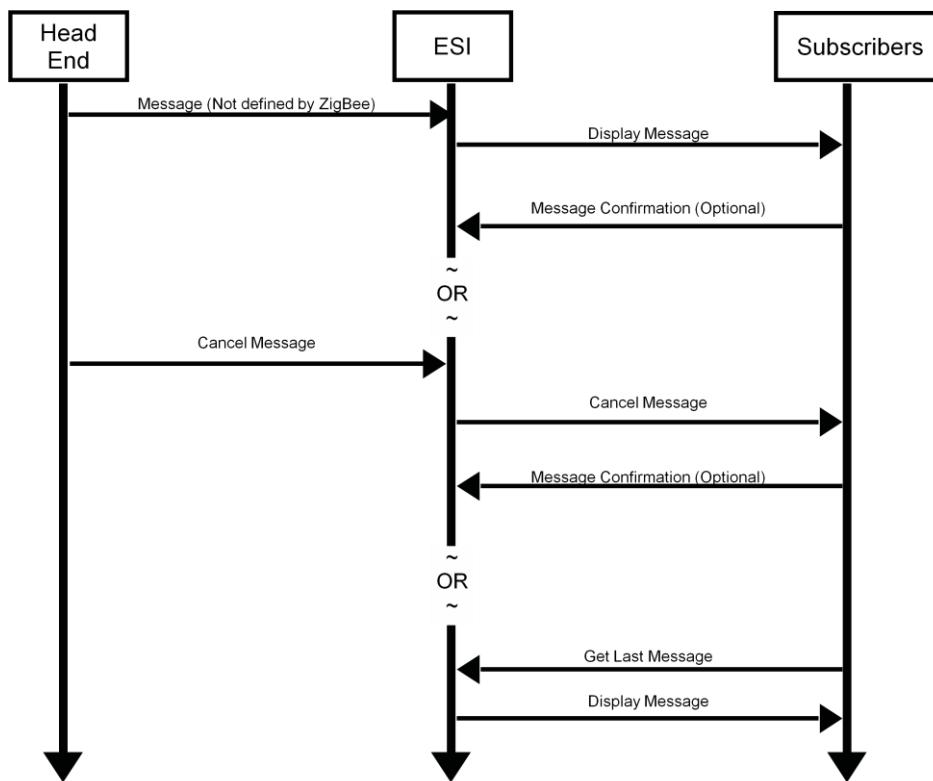
9880 A ZCL Default response with status NOT_FOUND shall be returned if there is no *Cancel All*
9881 *Messages* command available that satisfies the requested criteria.

9882

D.5.4 Application Guidelines

9884 For Server and Client transactions, please refer to Figure D-97.

9885



9886

9887

Figure D-97– Client/Server Message Command Exchanges

9888

9889

9890

9891 D.6 Tunneling Cluster

Note: The optional support for flow control within the cluster in this revision of this specification is provisional and not certifiable. This feature set may change before reaching certifiable status in a future revision of this specification.

9895 D.6.1 Overview

9896 The tunneling cluster provides an interface for tunneling protocols. It is comprised of commands
9897 and attributes required to transport any existing metering communication protocol within
9898 the payload of standard ZigBee frames (including the handling of issues such as addressing,
9899 fragmentation and flow control). Examples for such protocols are DLMS/COSEM, IEC61107,
9900 ANSI C12, M-Bus, ClimateTalk etc.

9901 The tunneling foresees the roles of a server and a client taking part in the data exchange.
9902 Their roles are defined as follows:

- **Client:** Requests a tunnel from the server and closes the tunnel if it is no longer needed.
 - **Server:** Provides and manages tunnels to the clients.

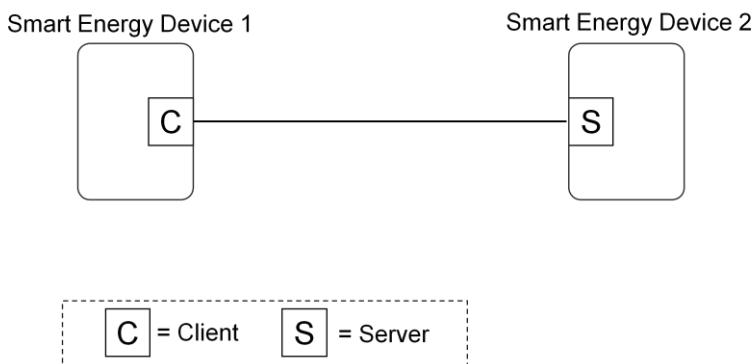


Figure D-98—A Client Requests a Tunnel From a Server to Exchange Complex Data in Both Directions

9908 The data exchange through the tunnel is symmetric. This means both client and server provide
9909 the commands to transfer data (*TransferData*). And both must make sure that only the partner to
9910 which the tunnel has been built up is granted read/write access to it (e.g. tunnel identifier protection
9911 through checking the MAC address).

9912 Sleepy devices either close the tunnel immediately after they have pushed their data through
9913 it, or leave it open in which case an attribute in the server (*CloseTunnelTimeout*) decides
9914 whether the tunnel is closed from the server side during the sleeping phase or not. It is
9915 recommended that battery-powered (sleepy) devices fulfil the role of the Tunneling cluster client
9916 (and therefore have control over when they request a tunnel from the server).

9917 If data is transferred to a non-existent or wrong tunnel identifier, the receiver generates an error
9918 message (*TransferDataError*).

9919 The server may support more than one tunneling protocol. The type of tunnel to be opened is a
9920 mandatory parameter (*ProtocolID*) of the tunnel request (*RequestTunnel*) that the client needs to
9921 send to the server in order to set up a new tunnel. The response from the server
9922 (*RequestTunnelResponse*) will contain a parameter with the status of the tunnel (*TunnelStatus*).
9923 If the tunnel request was successful, a unique identifier (*TunnelID*) is returned within the response.
9924 In an error case (e.g. the requested protocol is not supported) the status contains the type of
9925 error. The optional *GetSupportedTunnelProtocols* command provides a way to read out the
9926 supported protocols from the server. If the *GetSupportedTunnelProtocols* command is not
9927 supported then either the client knows the supported protocols a priori or it has to try several
9928 times using different *ProtocolIDs* until the server responds with the tunnel status *Success*.

9929 The tunneling cluster adds optional support for flow control to handle streaming protocols such
9930 as IEC61107. If implemented, flow control messages are provided to control the data flow and
9931 send acknowledges to data messages on application level. However, flow control is an optional
9932 feature and disabled per default. In the default case, the acknowledge messages
9933 (*AckTransferData*) must not be sent in order to reduce complexity and prevent from unneeded
9934 overhead.

9935 The following sequence describes a typical usage:

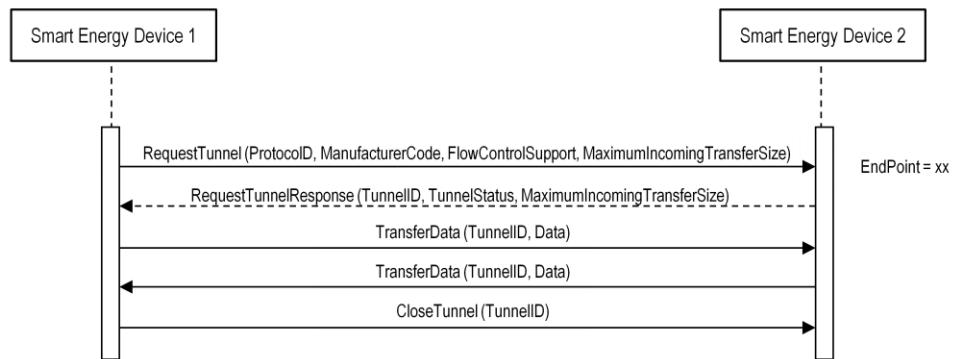
9936 1 The client issues a service discovery to find devices which support the tunneling server
9937 cluster. The discovery may either be directed to one device, if its address is known, or be a
9938 broadcast (*MatchSimpleDescriptor*).

9939 2 The response to the discovery from the server contains an endpoint number
9940 (*SimpleDescriptor*). Using this endpoint, the client directs a tunnel request to a given server.
9941 Together with the request, the client is required to provide an enumeration with the ID of the
9942 protocol that shall be tunneled. There is the possibility to request tunnels for manufacturer
9943 specific protocols. In this case, the *ProtocolID* has to be followed by a ZigBee
9944 *ManufacturerCode* to open the tunnel. An additional parameter for *FlowControlSupport*
9945 accompanies the request, together with an indication of the client's incoming buffer size
9946 (*RequestTunnel* (*ProtocolID*, *ManufacturerCode*, *FlowControlSupport*,
9947 *MaximumIncomingTransferSize*)).

9948 3 If the server supports the protocol, it allocates the required resources, assigns a tunnel
9949 identifier and returns the ID number within the response including an additional tunnel status
9950 that the command was successful and the server's incoming buffer size. If the command
9951 failed, the status contains the reason in form of an error code (*RequestTunnelResponse*
9952 (*TunnelID*, *TunnelStatus*, *MaximumIncomingTransferSize*)). The tunnel identifier number
9953 would then be invalid in this case.

- 9954 **4** Both server and client may exchange data (*TransferData(Data)*). In case the optional flow
 9955 control is utilized, each data transfer is acknowledged
 9956 (*AckTransferData(NumberOfOctetsLeft)*). Additionally, there is the possibility to stop
 9957 (*AckTransferData(0)*) and resume (*ReadyData(NumberOfOctetsLeft)*) the data transfer.
- 9958 **5** After the transfer has been successfully completed, the client closes the tunnel again freeing
 9959 the tunnel identifier in the server (*CloseTunnel(TunnelID)*). If not, the server closes the
 9960 tunnel by itself after *CloseTunnelTimeout* seconds.

9961 The following sequence diagrams show the client/server model and the typical usage of the
 9962 cluster without (Figure D-99) and with (Figure D-100) flow control.



9963 Without flow control

9964 **Figure D-99– SE Device 1 (Client) Requests a Tunnel From SE Device 2 (Server) to Transfer Data Without
 9965 Flow Control (Default)⁶**
 9966
 9967

⁶ CCB 1939



9968

9969 **Figure D-100—SE Device 1 (Client) Requests a Tunnel From SE Device 2 (Server) to Transfer Data With Flow
9970 Control⁷**
9971
99729973 **D.6.2 Server**9974 **D.6.2.1 Dependencies**9975 This cluster requires APS fragmentation [B3] to be implemented, with maximum transfer sizes
9976 defined by the device's negotiated input buffer sizes.9977 **D.6.2.2 Attributes**9978 **Table D-120—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

9979

⁷ CCB 1939

9980 D.6.2.2.1 **CloseTunnelTimeout Attribute**

9981 *CloseTunnelTimeout* defines the minimum number of seconds that the server waits on an
 9982 inactive tunnel before closing it on its own and freeing its resources (without waiting for the
 9983 *CloseTunnel* command from the client). Inactive means here that the timer is re-started with
 9984 each new reception of a command. 0x0000 is an invalid value.

9985

9986 D.6.2.3 Parameters

9987 The table below contains a summary of all parameters passed to or returned by the server
 9988 commands. These values are considered as parameters (and not attributes) in order to facilitate
 9989 the handling of the tunneling cluster for both the client and the server side. The parameters
 9990 cannot be read or written via ZCL global commands. The detailed description of these
 9991 parameters can be found in the according command sections of the document.

9992 **Table D-121– 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

9993

9994

9995

9996

9997 D.6.2.4 Commands Received

9998 Table D-122 lists cluster-specific commands received by the server.

9999

Table D-122– 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

10000

10001

10002 D.6.2.4.1 RequestTunnel Command

10003 *RequestTunnel* is the client command used to setup a tunnel association with the server. The
 10004 request payload specifies the protocol identifier for the requested tunnel, a manufacturer code
 10005 in case of proprietary protocols and the use of flow control for streaming protocols.

10006 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-101– *RequestTunnel* Command Payload

10007

10008

10009 D.6.2.4.1.2 Payload Details

10010 **ProtocolID:** An enumeration representing the identifier of the metering communication protocol
 10011 for which the tunnel is requested. Table D-123 lists the possible values for the *ProtocolID*.
 10012 The values above 199 may be used for manufacturer specific protocols.

10013

10014

10015

Table D-123– ProtocolID Enumerations

Values	Description

0	DLMS/COSEM (IEC 62056)
1	IEC 61107
2	ANSI C12
3	M-BUS
4	SML
5	ClimateTalk
6	GB-HRGP
7 to 199	Reserved for future growth
200 to 254	Manufacturer-defined protocols
255	Reserved

- 10016
- 10017 **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.
- 10022 **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).
- 10025 **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.
- 10027 D.6.2.4.1.3 When Generated
- 10028 Is never generated by the server.
- 10029 D.6.2.4.1.4 Effect on Receipt
- 10030 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.
- 10033 D.6.2.4.2 CloseTunnel Command
- 10034 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.
- 10038 D.6.2.4.2.1 Payload Format

Octets	2
--------	---

Data Type	Unsigned 16-bit integer
Field Name	TunnelID (M)

10039

Figure D-102– CloseTunnel Command Payload10040 D.6.2.4.2.2 Payload Details

10041 **TunnelID:** The identifier of the tunnel that shall be closed. It is the same number that has been
 10042 previously returned in the response to a *RequestTunnel* command. Valid numbers range between
 10043 0..65535 and must correspond to a tunnel that is still active and maintained by the server.

10044 D.6.2.4.2.3 When Generated

10045 This command is never generated by the server.

10046 D.6.2.4.2.4 Effect on Receipt

10047 In case the given *TunnelID* is correct, the server closes the tunnel and frees the resources. The
 10048 associated tunnel is no longer maintained. If the *TunnelID* value does not match an active tunnel
 10049 on the server, the server shall return a ZCL Default Response with status NOT_FOUND.

10050 D.6.2.4.3 TransferData Command

10051 Command that indicates (if received) that the client has sent data to the server. The data itself
 10052 is contained within the payload.

10053 D.6.2.4.3.1 Payload Format

Octets	2	Variable
Data Type	Unsigned 16-bit integer	Octets
Field Name	TunnelID (M)	Data (M)

10054

Figure D-103– TransferData Command Payload10055 D.6.2.4.3.2 Payload Details

10056 **TunnelID:** A number between 0..65535 that uniquely identifies the tunnel that has been
 10057 allocated in the server triggered through the *RequestTunnel* command. This ID must be used
 10058 to send data through the tunnel or passed with any commands concerning that specific
 10059 tunnel.

10060 **Data:** Octet containing the data to be transferred through the tunnel in the format of the
 10061 communication protocol for which the tunnel has been requested and opened. The payload
 10062 contains the assembled data exactly as it was sent by the client. Theoretically, its length is
 10063 solely limited through the fragmentation algorithm and the RX/TX transfer buffer sizes within the
 10064 communication partners. The content of the payload is up to the application sending the data. It
 10065 is neither guaranteed, that it contains a complete PDU nor is any other assumption on its

10066 internal format made. This is left up to the implementer of the specific protocol tunnel
 10067 behavior.

10068 D.6.2.4.3.3 When Generated

10069 Is generated whenever the server wants to tunnel protocol data to the client.

10070 D.6.2.4.3.4 Effect on Receipt

10071 Indicates that the server has received tunneled protocol data from the client.

10072 D.6.2.4.4 TransferDataError Command

10073 This command is generated by the receiver of a *TransferData* command if the tunnel status
 10074 indicates that something is wrong. There are three cases in which *TransferDataError* is sent:

- 10075 • The *TransferData* received contains a *TunnelID* that does not match to any of the active
 10076 tunnels of the receiving device. This could happen if a (sleeping) device sends a
 10077 *TransferData* command to a tunnel that has been closed by the server after the
 10078 *CloseTunnelTimeout*.
- 10079 • The *TransferData* received contains a proper *TunnelID* of an active tunnel, but the device
 10080 sending the data does not match to it.
- 10081 • The *TransferData* received contains more data than indicated by the
 10082 *MaximumIncomingTransferSize* of the receiving device.

10083 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)

10084 **Figure D-104– TransferDataError Command Payload**

10085 D.6.2.4.4.2 Payload Details

10086 **TunnelID:** A number between 0..65535 that uniquely identifies the tunnel that has been
 10087 allocated in the server triggered through the *RequestTunnel* command. This ID must be used
 10088 for the data transfer through the tunnel or passed with any commands concerning that specific
 10089 tunnel.

10090 **TransferDataStatus:** The *TransferDataStatus* parameter indicates the error that occurred
 10091 within the receiver after the last *TransferData* command.

10092 The *TransferDataStatus* values are shown in Table D-124.

10093

Table D-124– 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.

10094

10095 D.6.2.4.4.3 When Generated10096 Is generated if the server wants to tell the client that there was something wrong with the last
10097 *TransferData* command.10098 D.6.2.4.4.4 Effect on Receipt10099 Indicates that the client wants to tell the server that there was something wrong with the last
10100 *TransferData* command.10101 D.6.2.4.5 AckTransferData Command10102 Command sent in response to each *TransferData* command in case – and only in case – flow
10103 control has been requested by the client in the *TunnelRequest* command and is supported
10104 by both tunnel endpoints. The response payload indicates the number of octets that may still be
10105 received by the receiver.10106 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-105– AckTransferData Command Payload10108 D.6.2.4.5.2 Payload Details10109 **TunnelID:** A number between 0..65535 that uniquely identifies the tunnel that has been
10110 allocated in the server triggered through the *RequestTunnel* command. This ID must be used
10111 for the data transfer through the tunnel or passed with any commands concerning that specific
10112 tunnel.

10113 **NumberOfBytesLeft:** Indicates the number of bytes that may still be received by the initiator of
 10114 this command (receiver). It is most likely the remaining size of the buffer holding the data that is
 10115 sent over *TransferData*. As an example: A value of 150 indicates that the next *TransferData*
 10116 command must not contain more than 150 bytes of payload or data will get lost. A value of 0
 10117 indicates that there is no more space left in the receiver and the sender should completely stop
 10118 sending data. After the reception of a *ReadyData* command, the sender may continue its data
 10119 transfer.

10120 D.6.2.4.5.3 When Generated

10121 If flow control is on, the command is issued by the server to inform the client that the last
 10122 *TransferData* command has been successfully received and how much space is left to receive
 10123 further data.

10124 D.6.2.4.5.4 Effect on Receipt

10125 If flow control is on, the reception of this command indicates that the client wants to inform the
 10126 server that the last *TransferData* command has been successfully received and how much space
 10127 is left to receive further data.

10128 D.6.2.4.6 ReadyData Command

10129 The *ReadyData* command is generated – after a receiver had to stop the dataflow using the
 10130 *AckTransferData(0)* command – to indicate that the device is now ready to continue receiving
 10131 data. The parameter *NumberOfOctetsLeft* gives a hint on how much space is left for the next
 10132 data transfer. The *ReadyData* command is only issued if flow control is enabled.

10133 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)

10134 Figure D-106– *ReadyData* Command Payload

10135 D.6.2.4.6.2 Payload Details

10136 **TunnelID:** A number between 0..65535 that uniquely identifies the tunnel that has been
 10137 allocated in the server triggered through the *RequestTunnel* command. This ID must be used
 10138 for the data transfer through the tunnel or passed with any commands concerning that specific
 10139 tunnel.

10140 **NumberOfOctetsLeft:** Indicates the number of octets that may be received by the initiator of
 10141 this command (receiver). It is most likely the remaining size of the buffer holding the data that is
 10142 sent over *TransferData*. As an example: A value of 150 indicates that the next *TransferData*
 10143 command must not contain more than 150 bytes of payload or data will get lost. The value must

10144 be larger than 0. As for its exact value, it is up to the implementer of the cluster to decide
 10145 what flow control algorithm shall be applied.

10146 D.6.2.4.6.3 When Generated

10147 If generated by the server, this command informs the client that it may now continue to send and
 10148 how much space is left within the server to receive further data.

10149 D.6.2.4.6.4 Effect on Receipt

10150 If received by the server, this command informs the server that it may now continue to send and
 10151 how much space is left within the client to receive further data.

10152 D.6.2.4.7 **Get Supported Tunnel Protocols Command**

10153 *Get Supported Tunnel Protocols* is the client command used to determine the tunnel protocols
 10154 supported on another device.

10155 D.6.2.4.7.1 Payload Format

Octets	1
Data Type	Unsigned 8-bit Integer
Field Name	Protocol Offset

10156 **Figure D-107– *Get Supported Tunnel Protocols* Command Payload**

10157 D.6.2.4.7.2 Payload Details

10158 **Protocol Offset:** Where there are more protocols supported than can be returned in a single
Supported Tunnel Protocols Response command, this field allows an offset to be specified on
 10159 subsequent *Get Supported Tunnel Protocols* commands. An offset of zero (0x00) should be used
 10160 for an initial (or only) *Get Supported Tunnel Protocols* command (indicating that the returned list
 10161 of protocols should commence with first available protocol). As a further example, if 10 protocols
 10162 had previously been returned, the next *Get Supported Tunnel Protocols* command should use an
 10163 offset of 10 (0x0A) to indicate the 11th available protocol should be the first returned in the
 10164 next response.
 10165

10166 D.6.2.4.7.3 Effect on Receipt

10167 On receipt of this command, a device will respond with a *Supported Tunnel Protocols*
 10168 *Response* command, indicating the tunnel protocols it supports (see sub-clause D.6.2.5.6 for
 10169 further details).

10170 **D.6.2.5 Commands Generated**

10171 Table D-125 lists commands that are generated by the server.

10172 **Table D-125– 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

10173
10174

10175 D.6.2.5.1 **RequestTunnelResponse Command**

10176 *RequestTunnelResponse* is sent by the server in response to a *RequestTunnel* command
 10177 previously received from the client. The response contains the status of the *RequestTunnel*
 10178 command and a tunnel identifier corresponding to the tunnel that has been set-up in the server
 10179 in case of success.

10180 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)

10181 **Figure D-108– *RequestTunnelResponse* Command Payload**

10182 D.6.2.5.1.2 **Payload Details**

10183 **TunnelID:** A number between 0..65535 that uniquely identifies the tunnel that has been
 10184 allocated in the server triggered through the *RequestTunnel* command. This ID must now be
 10185 used to send data through this tunnel (*TunnelID*, *TransferData*) and is also required to
 10186 close the tunnel again (*CloseTunnel*). If the command has failed, the *TunnelStatus* contains the
 10187 reason of the error and the *TunnelID* is set to 0xFFFF.

10188 **TunnelStatus:** The *TunnelStatus* parameter indicates the server's internal status after the
 10189 execution of a *RequestTunnel* command.

10190 The *TunnelStatus* values are shown in Table D-126.

10191 **Table D-126– TunnelStatus Values**

Value	Description	Remarks
0x00	Success	The tunnel has been opened and may now be used to transfer data in both directions.
0x01	Busy	The server is busy and cannot create a new tunnel at the moment. The client may try again after a recommended timeout of 3 minutes.
0x02	No more tunnel IDs	The server has no more resources to setup requested tunnel. Clients should close any open tunnels before retrying.
0x03	Protocol not supported	The server does not support the protocol that has been requested in the ProtocolID parameter of the <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.

10192
 10193 **MaximumIncomingTransferSize:** A value that defines the size, in octets, of the maximum data packet that can be transferred to the server in the payload of a single *TransferData* command.
 10194

10195 D.6.2.5.1.3 When Generated

10196 Is generated in reply to a *RequestTunnel* command to inform the client about the result of the request.
 10197

10198 D.6.2.5.1.4 Effect on Receipt

10199 Should never be received by the server.

10200 D.6.2.5.2 TransferData Command

10201 Command that transfers data from server to the client. The data itself has to be placed within the payload.
 10202

10203 D.6.2.5.2.1 Payload Format

Octets	2	Variable
Data Type	Unsigned 16-bit integer	Octets
Field Name	TunnelID (M)	Data (M)

10204 Figure D-109– *TransferData* Command Payload

10205 D.6.2.5.2.2 Payload Details

10206 **TunnelID:** A number between 0..65535 that uniquely identifies the tunnel that has been allocated in the server triggered through the *RequestTunnel* command. This ID must be used
 10207

10208 for the data transfer through the tunnel or passed with any commands concerning that specific
 10209 tunnel.

10210 **Data:** Octets containing the data to be transferred through the tunnel in the format of the
 10211 communication protocol for which the tunnel has been requested and opened. The payload
 10212 containing the assembled data exactly as it has been sent away by the client. Theoretically, its
 10213 length is solely limited through the fragmentation algorithm and the RX/TX transfer buffer sizes
 10214 within the communication partners. The content of the payload is up to the application
 10215 sending the data. It is not guaranteed that it contains a complete PDU, nor is any assumption to
 10216 be made on its internal format (which is left up to the implementer of the specific tunnel
 10217 protocol).

10218 D.6.2.5.2.3 When Generated

10219 Is generated when the server wants to tunnel protocol data to the client.

10220 D.6.2.5.2.4 Effect on Receipt

10221 Indicates that the server has received tunneled protocol data from the client.

10222 D.6.2.5.3 **TransferDataError Command**

10223 See sub-clause D.6.2.4.4.

10224 D.6.2.5.4 **AckTransferData Command**

10225 See sub-clause D.6.2.4.5.

10226 D.6.2.5.5 **ReadyData Command**

10227 See sub-clause D.6.2.4.6.

10228 D.6.2.5.6 **Supported Tunnel Protocols Response Command**

10229 *Supported Tunnel Protocols Response* is sent in response to a *Get Supported Tunnel*
 10230 *Protocols* command previously received. The response contains a list of tunnel protocols
 10231 supported by the device; the payload of the response should be capable of holding up to 16
 10232 protocols.

10233

10234 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

10235 **Figure D-110– Supported Tunnel Protocols Response Command Payload**

10236 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

10237 **Figure D-111– Supported Tunnel Protocols Response Command Protocol Fields**

10238 D.6.2.5.6.2 Payload Details

10239 **Protocol List Complete:** The Protocol List Complete field is a Boolean; a value of 0 indicates
10240 that there are more supported protocols available (if more than 16 protocols are supported). A
10241 value of 1 indicates that the list of supported protocols is complete.

10242 **Protocol Count:** The number of Protocol fields contained in the response.

10243 **Manufacturer Code:** A code that is allocated by the ZigBee Alliance, relating the manufacturer
10244 to a device and - for tunneling - a manufacturer specific protocol. A value of 0xFFFF indicates a
10245 standard (i.e. non- manufacturer specific) protocol

10246 **Protocol ID:** An enumeration representing the identifier of the metering communication protocol
10247 for the supported tunnel. Table D-123 lists the possible values for standard protocols

10248 D.6.2.5.6.3 When Generated

10249 Is generated in reply to a *Get Supported Tunnel Protocols* command. to indicate the tunnel
10250 protocols supported by the device

10251 D.6.2.5.7 **TunnelClosureNotification Command**

10252 *TunnelClosureNotification* is sent by the server to indicate that a tunnel has been closed due to
10253 expiration of a *CloseTunnelTimeout*.

10254

10255 D.6.2.5.7.1 Payload Format

Octets	2
Data Type	Unsigned 16-bit Integer
Field Name	TunnelID (M)

10256 **Figure D-112– TunnelClosureNotification Command Payload**

10257 D.6.2.5.7.2 Payload Details

10258 **TunnelID:** The identifier of the tunnel that has been closed. It is the same number that has been
10259 previously returned in the response to a *RequestTunnel* command. Valid numbers range

10260 between 0..65535 and must correspond to a tunnel that was still active and maintained by the
10261 server.

10262 D.6.2.5.7.3 When Generated

10263 The command is sent by a server when a tunnel is closed due to expiration of
10264 *CloseTunnelTimeout*. It is sent unicast to the client that had originally requested that tunnel.

10265

10266 **D.6.3 Client**

10267 **D.6.3.1 Dependencies**

10268 This cluster requires APS fragmentation [B3] to be implemented, with maximum transfer sizes
10269 defined by the device's negotiated input buffer sizes.

10270 **D.6.3.2 Attributes**

10271 The client has no attributes.

10272 **D.6.3.3 Commands Received**

10273 The client receives the cluster-specific response commands detailed in D.6.2.5.

10274 **D.6.3.4 Commands Generated**

10275 The client generates the cluster-specific commands detailed in D.6.2.4, as required by the
10276 application.

10277

10278

10279 **D.7 Prepayment Cluster**

10280 *Note: The Prepayment Cluster description in this revision of this specification is provisional
10281 and not certifiable. This feature set may change before reaching certifiable status in a future
10282 revision of this specification.*

10283 **D.7.1 Overview**

10284 The Prepayment Cluster provides the facility to pass messages relating to the accounting
10285 functionality of a meter between devices on the HAN. It allows for the implementation of a
10286 system conforming to the set of standards relating to Payment Electricity Meters (IEC 62055)
10287 and also for the case where the accounting function is remote from the meter. Prepayment is
10288 used in situations where the supply of a service may be interrupted or enabled under the control
10289 of the meter or system in relation to a payment tariff. The accounting process may be within
10290 the meter or elsewhere in the system. The amount of available credit is decremented as the
10291 service is consumed and is incremented through payments made by the consumer. Such a
10292 system allows the consumer to better manage their energy consumption and reduces the risk
10293 of bad debt owing to the supplier.

10294 In the case where the accounting process resides within the meter, credit updates are sent to the
10295 meter from the ESI. Such messages are out of scope of this cluster. The cluster allows credit
10296 status to be made available to other devices on the HAN for example to enable the
10297 consumers to view their status on an IHD. It also allows them to select emergency credit if
10298 running low and also, where local markets allow, restoring their supply remotely from within the
10299 HAN.

10300 In the case where the accounting process resides in the head end (Central Wallet scheme), the
10301 metering system provides usage information to the head end for it to calculate the state of
10302 available credit in the consumer's account. The head end will pass down to the metering system
10303 data that will be of use to the consumer, for distribution on the HAN. The head end will also
10304 send commands to interrupt or restore the supply depending on the state of the account.

10305 In either case, there will be the need to display credit status and this may be in monetary terms
10306 or in energy terms. If running in monetary mode, the units of measure will be defined in the
10307 Price Cluster, if in energy terms, the unit of measure will be defined in the Metering Cluster.

10308



C = Client S = Server

Figure D-113– Prepay Cluster Client Server Example

10309
10310

10311

10312 **D.7.2 Server**

10313 **D.7.2.1 Dependencies**

- 10314 • Support for ZCL Data Types
- 10315 • Events carried using this cluster include a timestamp with the assumption that target devices
10316 maintain a real time clock. Devices can acquire and synchronize their internal clocks via the
10317 ZCL Time server.
- 10318 • Use of the Price cluster is Mandatory when using the Prepayment cluster in Currency mode.
- 10319 • The Calendar cluster shall be used to set up the Friendly Credit period that the prepayment
10320 meter shall use (see Annex D.9 for further details).
- 10321 • Use of the Metering cluster is Mandatory when using the Prepayment cluster in any mode.
- 10322 • Use of the Device Management cluster is mandatory when using the disconnection function
10323 within the Prepayment cluster.

10324 **D.7.2.2 Attributes**

10325 For convenience, the attributes defined in this specification are arranged into sets of related
10326 attributes; each set can contain up to 256 attributes. Attribute identifiers are encoded such that the
10327 most significant Octet specifies the attribute set and the least significant Octet specifies the
10328 attribute within the set. The currently defined attribute sets are listed in the following Table
10329 D-127.

10330

Table D-127– 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

10332

10333 D.7.2.2.1 **Prepayment Information Attribute Set**10334 The following set of attributes provides access to the standard information relating to a
10335 Prepayment meter.

10336

Table D-128– 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	CreditRemainingTimeStamp	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 0xFFFFFFFFFFFF	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					

10337

10338 D.7.2.2.1.1 PaymentControl Configuration Attribute10339 The *PaymentControlConfiguration* attribute represents the payment mechanisms currently
10340 enabled within the Metering Device. Bit encoding of this field is outlined in Table D-129.

10341

Table D-129– 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

10342

10343 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	1	1	1	0	1	0	0	1	X	X	X	

Mode of operation	Description	Bits									
		0	1	2	3	4	5	6	7	8	9
	remote supply interruption commands										

- 10344
- 10345 **Disconnection Enabled:** Indicates whether the metering device is to disconnect the energy supply on expiry of available credit.
- 10346
- 10347 **Prepayment Enabled:** Indicates if the meter is a ‘prepayment’ meter; if this value is 0, the meter is considered to be a ‘credit’ meter.
- 10348
- 10349 **Credit Management Enabled:** Indicates whether the metering device should manage accounting functionality according to available tariff information.
- 10350
- 10351 **Credit Display Enabled:** Indicates whether the metering device should display the credit status.
- 10352
- 10353 **Account Base:** Indicates whether the metering device is running in Monetary (0) or Unit based (1) units. If Monetary based, the unit of measure is defined in the Price cluster, if Unit based, the unit of measure is defined in the Metering cluster
- 10354
- 10355
- 10356 **Contactor Fitted:** Indicates whether the metering device is fitted with a Contactor i.e. is capable of disconnecting the energy supply.
- 10357
- 10358 **Standing Charge Configuration:** Indicates whether the standing charge collection is halted when the prepaid credit is exhausted.
- 10359
- 10360 **Emergency Standing Charge Configuration:** Indicates whether the standing charge collection is halted when the device is in Emergency Credit mode.
- 10361
- 10362 **Debt Configuration:** Indicates whether the debt collection is halted when the prepaid credit is exhausted.
- 10363
- 10364 **Emergency Debt Configuration:** Indicates whether the debt is collected when the device is in Emergency Credit mode.
- 10365
- 10366 D.7.2.2.1.2 Credit Remaining Attribute
- 10367 The *Credit Remaining* attribute represents the amount of credit remaining on the Metering Device. If Monetary-based, this attribute is measured in a base unit of *Currency* with the decimal point located as indicated by the *Trailing Digits* field, as defined in the Price cluster. If Unit-based, the unit of measure is as defined in the Metering cluster (see sub-clause D.3.2.2.4.1).
- 10368
- 10369
- 10370

10371 D.7.2.2.1.3 Emergency Credit Remaining Attribute

10372 The *Emergency Credit Remaining* attribute represents the amount of Emergency Credit still
 10373 available on the Metering Device. If Monetary-based, this attribute is measured in a base unit of
 10374 *Currency* with the decimal point located as indicated by the *Trailing Digits* field, as defined in
 10375 the Price cluster. If Unit-based, the unit of measure is as defined in the Metering cluster (see sub-
 10376 clause D.3.2.2.4.1).

10377 D.7.2.2.1.4 Credit Status Attribute

10378 The *Credit Status* attribute represents the current status of credit within the Metering Device.
 10379 Bit encoding of this field is outlined in Table D-130. Explanation of the use of this attribute can
 10380 be found in section D.7.4.1.

10381 **Table D-130—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

10382

10383 D.7.2.2.1.5 CreditRemainingTimeStamp Attribute

10384 The UTC time at which the *Credit Remaining* attribute was last populated.

10385 D.7.2.2.1.6 AccumulatedDebt Attribute

10386 The *AccumulatedDebt* attribute represents the total amount of debt remaining on the Metering
 10387 Device. This attribute is always Monetary based and, as such, this attribute is measured in a base
 10388 unit of *Currency* with the decimal point located as indicated by the *Trailing Digits* field, as
 10389 defined in the Price cluster.

10390 D.7.2.2.1.7 OverallDebtCap Attribute

10391 The *OverallDebtCap* attribute represents the total amount of debt that can be taken from top-ups
 10392 (in the case of multiple instantiated top-up based debts on the Metering Device). This attribute is
 10393 configured to the required limit per unit time (fixed globally in the application at one week) that
 10394 the consumer pays off against their debts. This attribute is always a monetary value, and as such
 10395 this attribute is measured in a base unit of *Currency* with the decimal point located as indicated
 10396 by the *Trailing Digits* field, as defined in the Price cluster.

10397 As an example, a consumer has a single Percentage Based debt in operation, with a collection
 10398 rate of 20% and an *OverallDebtCap* of £5 per week. He buys £5 credit every day. Table D-131
 10399 shows the resultant allocation of the amounts purchased:

10400

Table D-131– 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

10401
 10402 Once the cap value has been reached during a week then no further amounts are deducted from
 10403 the purchases.

10404 As an extension of the example, if the customer purchased £50 credit on the Monday, the meter
 10405 would take £5 (not £10) and would also not take any further debt payments from any other
 10406 purchases made in the same week.

10407 D.7.2.2.1.8 EmergencyCreditLimit/Allowance Attribute

10408 The *EmergencyCreditLimit/Allowance* attribute may be updated by the utility company. This is
 10409 the amount of Emergency Credit available to loan to the consumer when the remaining balance
 10410 goes below the low credit threshold. If Monetary based, then this attribute is measured in base
 10411 unit of *Currency* with the decimal point located as indicated by the *Trailing Digits* field, as
 10412 defined in the Price cluster. If Unit based, the unit of measure is as defined in the Metering
 10413 cluster (see sub-clause D.3.2.2.4.1).

10414 D.7.2.2.1.9 EmergencyCreditThreshold Attribute

10415 When credit (or emergency credit) falls below this threshold, an alarm is raised to warn the
 10416 consumer of imminent supply interruption and, if available, to offer Emergency Credit. If
 10417 Monetary based, the unit of measure is the same as that defined in the Price cluster. If Unit based,
 10418 the unit of measure is as defined in the Metering cluster (see sub-clause D.3.2.2.4.1).

10419 D.7.2.2.1.10 TotalCreditAdded Attribute

10420 An unsigned 48-bit integer value indicating running total of credit topped up to date. If Monetary
 10421 based, this attribute is measured in a base unit of *Currency* with the decimal point located as
 10422 indicated by the *Trailing Digits* field, as defined in the Price cluster. If Unit based, the unit of
 10423 measure is as defined in the Metering cluster (see sub-clause D.3.2.2.4.1). At change of Tenant or
 10424 Supplier, this attribute shall be reset to zero.

10425 D.7.2.2.1.11 MaxCreditLimit Attribute

10426 An unsigned 32-bit integer value indicating the maximum credit balance allowed on a meter.
10427 Any further top-up amount that will cause the meter to exceed this limit will be rejected. This
10428 value can be stated in currency (as per the Price cluster) or in units (unit of measure will be
10429 defined in the Metering cluster) depending on the Prepayment mode of operation defined in
10430 section D.7.2.2.1.1 (*Payment Control Configuration* attribute). A value of 0xFFFFFFFF shall
10431 indicate that this limit is disabled and that all further top-ups should be permitted.

10432 D.7.2.2.1.12 MaxCreditPerTopUp Attribute

10433 An unsigned 32-bit integer value indicating the maximum credit per top-up. Any single top-up
10434 greater than this threshold will cause the meter to reject the top-up. This value can be stated in
10435 currency (as per the Price cluster) or in units (unit of measure will be defined in the Metering
10436 cluster) depending on the Prepayment mode of operation defined in section D.7.2.2.1.1 (*Payment
10437 Control Configuration* attribute). A value of 0xFFFFFFFF shall indicate that this parameter is
10438 disabled and that there should be no limit on the amount of allowed credit in a top-up.

10439 D.7.2.2.1.13 FriendlyCreditWarning Attribute

10440 An unsigned 8-bit integer value indicating the amount of time, in minutes, before the *Friendly
10441 Credit Period End Warning* alarm flag is triggered. The default value is 10 mins before the
10442 currently active Friendly Credit period is due to end.

10443 D.7.2.2.1.14 LowCreditWarningLevel Attribute

10444 An unsigned 32 bit integer that defines the **utility** low credit value below which the Low Credit
10445 warning should sound. The Low Credit warning shall be triggered when the value between the
10446 remaining credit and the disconnection point falls below this value. Falling below this value shall
10447 trigger the Low Credit warning alert within this cluster. The value is in a base unit of *Currency*
10448 (as per the Price cluster) or in Units (as per the Metering cluster). The attribute is set from the
10449 backhaul connection.

10450 D.7.2.2.1.15 IHDLowCreditWarningLevel Attribute

10451 An unsigned 32 bit integer that is defined by the **consumer** for a low credit value below which a
10452 Low Credit warning should sound. The Low Credit warning shall be triggered when the value
10453 between the remaining credit and the disconnection point falls below this value. This shall not
10454 trigger the Low Credit warning alert within this cluster. The value is in a base unit of *Currency*
10455 (as per the Price cluster) or in Units (as per the Metering cluster).

10456 D.7.2.2.1.16 InterruptSuspendTime Attribute

10457 When the end of a configured non-disconnect period is reached and the supply is to be
10458 interrupted due to insufficient credit being available, the meter will provide visual and audible
10459 alerts and the interruption will be suspended for a further period of minutes defined by this
10460 attribute. If no payments are applied to the meter during this period, or if insufficient credit is

- 10461 added, then, at the end of this period, an alert will be provided and the supply will then be
10462 interrupted.
- 10463 D.7.2.2.1.17 RemainingFriendlyCreditTime Attribute
10464 An unsigned 16-bit integer value indicating the amount of time remaining, in minutes, in a
10465 currently active Friendly Credit period. A value of zero shall indicate that no period is currently
10466 active (i.e. 0 = expired/no minutes left).
- 10467 D.7.2.2.1.18 NextFriendlyCreditPeriod Attribute
10468 The UTC time at which the next Friendly Credit period is due to commence.
- 10469 D.7.2.2.1.19 CutOffValue Attribute
10470 This attribute is a signed 32 bit integer that shall either be zero or a negative value (in all known
10471 cases). This allowance is measured in base unit of *Currency* with the decimal point located as
10472 indicated by the *Trailing Digits* field, as defined in the Price cluster. If Unit based, the unit of
10473 measure is as defined in the Metering cluster (see sub-clause D.3.2.2.4.1).
- 10474 This attribute represents a threshold relating to the absolute value of the *CreditRemaining*
10475 attribute, that when reached (when credit is decrementing) causes the supply of service to be
10476 disconnected. There can be several types of credit within a payment metering system of which
10477 there are 2 specified in this specification (*Credit* and *EmergencyCredit*). The *CreditRemaining*
10478 attribute shall contain the net worth of a consumers account within the meter, consolidating all
10479 active credit types (both *Credit* and *EmergencyCredit* if in use). As *EmergencyCredit* is
10480 effectively a loan from the supplier it becomes a liability once it is used, and when it is exhausted
10481 will force the *RemainingCredit* to a negative value. There are a number of other factors that can
10482 affect the way a prepayment meter works and which values are displayed to the end consumer.
10483 However, when a meter's *EmergencyCredit* has run out, the *CreditRemaining* value shall contain
10484 the total liability of the consumer (that he is required to pay before *EmergencyCredit* shall be
10485 available again) as a negative value.
- 10486 D.7.2.2.1.20 TokenCarrierId Attribute
10487 The *TokenCarrierId* attribute provides a method for utilities to publish the payment card number
10488 that is used with this meter set. The *TokenCarrierId* attribute is a ZCL Octet String capable of
10489 storing a 20 character string (the first Octet indicates length) encoded in the UTF-8 format. The
10490 *TokenCarrierId* attribute represents the current active value for the property.
- 10491 D.7.2.2.2 Top-up Attribute Set
- 10492 The following set of attributes provides access to previous successful credit *top-ups* on a
10493 prepayment meter. #1 is the most recent, based on time.
- 10494

Table D-132– 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	-0x7FFFFFFFFFFF to +0x7FFFFFFFFF	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					

10495

10496 D.7.2.2.1 Top up Date/Time Attribute

10497 The *Top up Date/Time* attribute represents the time that the credit was topped up on the
 10498 Metering Device. There are five records containing this attribute, one for each of the last five
 10499 top-ups.

10500 D.7.2.2.2 Top up Amount Attribute

10501 The *Top up Amount* attribute represents the amount of credit that was added to the Metering
 10502 Device during the top up. If Monetary-based, this attribute is measured in a base unit of
 10503 *Currency* with the decimal point located as indicated by the *Trailing Digits* field, as defined
 10504 in the Price cluster. If Unit-based, the unit of measure is as defined in the Metering cluster
 10505 (see sub-clause D.3.2.2.4.1). There are five records containing this attribute, one for each of the
 10506 last five top-ups.

10507 D.7.2.2.3 Originating Device Attribute

10508 The *Originating Device* attribute represents the SE device that was the source of the top-up
 10509 command. The enumerated values of this field are outlined in Table D-145. There are five
 10510 records containing this attribute, one for each of the last five top-ups.

10511 D.7.2.2.4 Top up Code Attribute

10512 The *Top up Code* attribute represents any encrypted number that was used to apply the credit to
 10513 the meter; the octet string shall be as it was received, i.e. not decoded. There are five records
 10514 containing this attribute, one for each of the last five top-ups.

10515 D.7.2.2.3 Debt Attribute Set

10516 The following set of attributes provides access to information on debt held on a Prepayment
 10517 meter.

10518

Table D-133– 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					

10519

10520 D.7.2.2.3.1 DebtLabel#N Attribute10521 The *DebtLabel#n* attribute provides a method for utilities to assign a name to a particular type of
10522 debt. The *DebtLabel#n* attribute is a ZCL Octet String field capable of storing a 12 character

10523 string (the first Octet indicates length) encoded in the UTF-8 format. This applies to all debt
 10524 recovery methods.

10525 D.7.2.2.3.2 [DebtAmount#N Attribute](#)

10526 An unsigned 32-bit field to denote the amount of Debt remaining on the Metering Device. This
 10527 parameter shall be measured in base unit of *Currency* with the decimal point located as indicated
 10528 by the *Trailing Digits* field, as defined in the Price Cluster.

10529 D.7.2.2.3.3 [DebtRecoveryMethod#N Attribute](#)

10530 An enumerated attribute denoting the debt recovery method used for this debt type. The
 10531 enumerated values for this field are outlined in Table D-134 (Time based, Percentage based and
 10532 Catch-Up based). This applies to all debt recovery methods.

10533

Table D-134– Debt Recovery Method Enumerations

Enumerated Value	Recovery Method
0x00	Time Based
0x01	Percentage Based
0x02	Catch-Up Based (Fixed Period)
0x03 – 0xFF	Reserved

10534

10535 D.7.2.2.3.4 [DebtRecoveryStartTime#N Attribute](#)

10536 A UTC Time field to denote the time at which the debt collection should start. This applies to all
 10537 debt recovery methods.

10538 D.7.2.2.3.5 [DebtRecoveryCollectionTime#N Attribute](#)

10539 An unsigned 16-bit field denoting the time of day when the debt collection takes place. It is
 10540 encoded as the number of minutes after midnight and has a valid range 0 .. 1440 with a default
 10541 value of 0. This applies to all debt recovery methods.

10542 D.7.2.2.3.6 [DebtRecoveryFrequency#N Attribute](#)

10543 The *DebtRecoveryFrequency#N* attribute represents the period over which each
 10544 *DebtRecoveryAmount#N* is recovered. The enumerated values of this field are outlined in Table
 10545 D-135.

10546

Table D-135–Recovery Frequency Field Enumerations

Enumerated Value	Recovery Period
0x00	Per Hour
0x01	Per Day
0x02	Per Week

0x03	Per Month
0x04	Per Quarter

10547

10548 D.7.2.2.3.7 DebtRecoveryAmount#N Attribute

10549 The *DebtRecoveryAmount#N* attribute represents the amount of Debt recovered each period
 10550 specified by *DebtRecoveryFrequency#N*, measured in base unit of *Currency* with the decimal
 10551 point located as indicated by the *Trailing Digits* field, as defined in the Price Cluster. This
 10552 attribute only applies to Time based and Catch-Up based debt recovery. A value of 0 indicates
 10553 not used.

10554

10555 D.7.2.2.3.8 DebtRecoveryTopUpPercentage#N Attribute

10556 An unsigned 16-bit field used in Percentage based recovery to denote the percentage from a top-
 10557 up amount to be deducted from the debt. For example, if the *DebtRecoveryTopUpPercentage#N*
 10558 is set to 10% and the customer topped up the device with 10 units of *Currency*, then 1 unit is
 10559 deducted from the amount being topped up and paid towards the debt recovery, i.e the device is
 10560 credited with only 9 units of currency. The percentage is always in the following format xxx.xx.
 10561 The default is 0.00% and maximum value is 100.00%.

10562 D.7.2.2.4 Supply Control Set

10563 The Supply Control functionality has been moved to the Metering cluster (see Annex D.3 for
 10564 further details).

10565

10566 D.7.2.2.5 Alarms Attribute Set

10567 The following set of attributes provides a means to control which prepayment alarms may be
 10568 generated from the meter.

10569 Table D-136– 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

10570

10571 D.7.2.2.5.1 Prepayment Alarm Status Attribute

10572 The *PrepaymentAlarmStatus* attribute provides indicators reflecting the current error conditions
 10573 found by the prepayment metering device. This attribute is a 16-bit field where when an
 10574 individual bit is set, an error or warning condition exists. The behaviour causing the setting or
 10575 resetting of each bit is device specific. In other words, the application within the prepayment
 10576 metering device will determine and control when these settings are either set or cleared. The ESI
 10577 should make alarms available to upstream systems, together with consumption data collected
 10578 from a battery operated meter.

10579 **Table D-137– 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	

10580

10581 D.7.2.2.5.2 Alarm Mask Attributes

10582 The Alarm Mask attributes of the Alarms Attribute Set specify whether each of the alarms listed
 10583 in the corresponding alarm group in Table D-138 through Table D-141 is enabled. When the bit
 10584 number corresponding to the alarm number (minus the group offset) is set to 1, the alarm is
 10585 enabled, else it is disabled. Bits not corresponding to a code in the respective table are reserved.

10586 D.7.2.2.5.3 Alarm Codes

10587 The alarm codes are organised in logical groups corresponding to the types of activity as listed
 10588 below. The three main alarm groups are: GenericAlarmMask, PrepaySwitchAlarmMask, and
 10589 PrepayEventAlarmMask.

10590 **Table D-138– Alarms Code Group**

Enumerated Alarm Codes	Alarm Condition
0x00 – 0x0F	PrePayGenericAlarmGroup
0x10 – 0x1F	PrepaySwitchAlarmGroup
0x20 – 0x4F	PrepayEventAlarmGroup
0x50 – 0xFF	Reserved

10591

10592 The Alarms that can be enabled/disabled in the PrepayGenericAlarmGroup are as follows:

10593

Table D-139– 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

10594

10595 The Alarms that can be enabled/disabled in the *PrepaySwitchAlarmGroup* are as follows:

10596

Table D-140– PrepaySwitchAlarmGroup

Enumerated Alarm Code	Alarm Condition
0x10	Supply ON
0x11	Supply ARM
0x12	Supply OFF
0x13	Disconnection Failure (Shut Off Mechanism Fail)
0x14	Disconnected due to Tamper Detected.

Enumerated Alarm Code	Alarm Condition
0x15	Disconnected due to Cut off Value.
0x16	Remote Disconnected.
0x17 – 0x1F	Reserved

10597

10598 The Alarms that can be enabled/disabled in the *PrepayEventAlarmGroup* are as follows:

10599

Table D-141– 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

10600

10601 D.7.2.2.6 **Historical Cost Consumption Information Set**

10602

Table D-142– 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	Consumption UnitofMeasurement	8-bit Enumeration	0x00 to 0xFF	Read Only	0x00	O
0x0502	CurrencyScalingFactor	8-bit Enumeration	0x00 to 0xFF	Read Only	-	O
0x0503	Currency	Unsigned 16-bit Integer	0x0000 to 0xFFFF	Read Only	-	O
0x0504-0x051B	Reserved					
0x051C	CurrentDay CostConsumptionDelivered	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x051D	CurrentDay CostConsumptionReceived	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x051E	PreviousDay CostConsumptionDelivered	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x051F	PreviousDay CostConsumptionReceived	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0520	PreviousDay2 CostConsumptionDelivered	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0521	PreviousDay2 CostConsumptionReceived	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0522	PreviousDay3 CostConsumptionDelivered	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0523	PreviousDay3 CostConsumptionReceived	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0524	PreviousDay4 CostConsumptionDelivered	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0525	PreviousDay4 CostConsumptionReceived	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0526	PreviousDay5 CostConsumptionDelivered	Unsigned 48-bit Integer	0x000000000000 to 0xFFFFFFFFFFFF	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 0xFFFFFFFFFFFF	Read Only	-	O
0x0558	PreviousMonth12 CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x0559	PreviousMonth12 CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x055A	PreviousMonth13 CostConsumptionDelivered	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x055B	PreviousMonth13 CostConsumptionReceived	Unsigned 48-bit Integer	0x0000000000 to 0xFFFFFFFFFFFF	Read Only	-	O
0x055C	Historical Freeze Time	Unsigned 16-bit Integer	0x0000 to 0x173C	Read Only	0x0000	O
0x055D- 0x05FF	Reserved					

10603

10604 D.7.2.2.6.1 HistoricalCostConsumptionFormatting Attribute

10605 *HistoricalCostConsumptionFormatting* provides a method to properly decipher the decimal point
 10606 location for the values found in the Historical Cost Consumption Set of attributes. The most
 10607 significant nibble indicates the number of digits to the left of the decimal point, the least
 10608 significant nibble the number of digits to the right.

10609 This attribute shall be used against the following attributes:

- 10610 • *CurrentDayCostConsumptionDelivered*
- 10611 • *CurrentDayCostConsumptionReceived*
- 10612 • *PreviousDayNCostConsumptionDelivered*
- 10613 • *PreviousDayNCostConsumptionReceived*
- 10614 • *CurrentWeekCostConsumptionDelivered*
- 10615 • *CurrentWeekCostConsumptionReceived*
- 10616 • *PreviousWeekNCostConsumptionDelivered*
- 10617 • *PreviousWeekNCostConsumptionReceived*
- 10618 • *CurrentMonthCostConsumptionDelivered*
- 10619 • *CurrentMonthCostConsumptionReceived*
- 10620 • *PreviousMonthNCostConsumptionDelivered*
- 10621 • *PreviousMonthNCostConsumptionReceived*

- 10622
- 10623 D.7.2.2.6.2 ConsumptionUnitofMeasurement Attribute
- 10624 *ConsumptionUnitofMeasurement* provides a label for the Energy, Gas, or Water being measured
10625 by the metering device. This attribute is an 8-bit enumerated field. The bit descriptions for this
10626 attribute are listed in Table D-26.
- 10627 This attribute shall be used against the following attributes:
- 10628 • *CurrentDayCostConsumptionDelivered*
- 10629 • *CurrentDayCostConsumptionReceived*
- 10630 • *PreviousDayNCostConsumptionDelivered*
- 10631 • *PreviousDayNCostConsumptionReceived*
- 10632 • *CurrentWeekCostConsumptionDelivered*
- 10633 • *CurrentWeekCostConsumptionReceived*
- 10634 • *PreviousWeekNCostConsumptionDelivered*
- 10635 • *PreviousWeekNCostConsumptionReceived*
- 10636 • *CurrentMonthCostConsumptionDelivered*
- 10637 • *CurrentMonthCostConsumptionReceived*
- 10638 • *PreviousMonthNCostConsumptionDelivered*
- 10639 • *PreviousMonthNCostConsumptionReceived*
- 10640
- 10641 D.7.2.2.6.3 CurrencyScalingFactor Attribute
- 10642 *CurrencyScalingFactor* provides a scaling factor for the *Currency* attribute for the Energy, Gas,
10643 or Water being measured by the metering device. This attribute is an 8-bit enumeration, the
10644 enumerated values for which are outlined in Table D-143. Note that this attribute will allow for a
10645 different resolution for historical values compared to values in the Price cluster.
- 10646 This attribute shall be used against the following attributes:
- 10647 • *CurrentDayCostConsumptionDelivered*
- 10648 • *CurrentDayCostConsumptionReceived*
- 10649 • *PreviousDayNCostConsumptionDelivered*
- 10650 • *PreviousDayNCostConsumptionReceived*
- 10651 • *CurrentWeekCostConsumptionDelivered*
- 10652 • *CurrentWeekCostConsumptionReceived*
- 10653 • *PreviousWeekNCostConsumptionDelivered*
- 10654 • *PreviousWeekNCostConsumptionReceived*
- 10655 • *CurrentMonthCostConsumptionDelivered*
- 10656 • *CurrentMonthCostConsumptionReceived*

- 10657 • *PreviousMonthNCostConsumptionDelivered*
 10658 • *PreviousMonthNCostConsumptionReceived*

10659
 10660 **Table D-143—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

10661
 10662

10663 **D.7.2.2.6.4 Currency Attribute**

10664 *Currency* provides the currency for the Energy, Gas, or Water being measured by the prepayment
 10665 device. This unsigned 16-bit value indicates the currency in which the following attributes are
 10666 represented:

- 10667 • *CurrentDayCostConsumptionDelivered*
 10668 • *CurrentDayCostConsumptionReceived*
 10669 • *PreviousDayNCostConsumptionDelivered*
 10670 • *PreviousDayNCostConsumptionReceived*
 10671 • *CurrentWeekCostConsumptionDelivered*
 10672 • *CurrentWeekCostConsumptionReceived*
 10673 • *PreviousWeekNCostConsumptionDelivered*
 10674 • *PreviousWeekNCostConsumptionReceived*
 10675 • *CurrentMonthCostConsumptionDelivered*
 10676 • *CurrentMonthCostConsumptionReceived*
 10677 • *PreviousMonthNCostConsumptionDelivered*
 10678 • *PreviousMonthNCostConsumptionReceived*

10680 D.7.2.2.6.5 CurrentDayCostConsumptionDelivered Attribute

10681 *CurrentDayCostConsumptionDelivered* represents the summed value of Energy, Gas, or Water
10682 delivered to the premises since the HFT. If optionally provided,
10683 *CurrentDayCostConsumptionDelivered* is updated continuously as new measurements are made.
10684 If the optional *Historical Freeze Time* attribute is not available, default to midnight local time.

10685 D.7.2.2.6.6 CurrentDayCostConsumptionReceived Attribute

10686 *CurrentDayCostConsumptionReceived* represents the summed value of Energy, Gas, or Water
10687 received from the premises since the HFT. If optionally provided,
10688 *CurrentDayCostConsumptionReceived* is updated continuously as new measurements are made.
10689 If the optional *Historical Freeze Time* attribute is not available, default to midnight local time.

10690 D.7.2.2.6.7 PreviousDayNCostConsumptionDelivered Attribute

10691 *PreviousDayNCostConsumptionDelivered* represents the summed value of Energy, Gas, or Water
10692 delivered to the premises within the previous 24 hour period starting at the HFT. If the
10693 optional *Historical Freeze Time* attribute is not available, default to midnight local time.

10694 D.7.2.2.6.8 PreviousDayNCostConsumptionReceived Attribute

10695 *PreviousDayNCostConsumptionReceived* represents the summed value of Energy, Gas, or Water
10696 received from the premises within the previous 24 hour period starting at the HFT. If the optional
10697 *Historical Freeze Time* attribute is not available, default to midnight local time.

10698 D.7.2.2.6.9 CurrentWeekCostConsumptionDelivered Attribute

10699 *CurrentWeekCostConsumptionDelivered* represents the summed value of Energy, Gas, or Water
10700 delivered to the premises since the HFT on Monday to the last HFT read. If optionally provided,
10701 *CurrentWeekCostConsumptionDelivered* is updated continuously as new measurements are
10702 made. If the optional *Historical Freeze Time* attribute is not available, default to midnight local
10703 time.

10704 D.7.2.2.6.10 CurrentWeekCostConsumptionReceived Attribute

10705 *CurrentWeekCostConsumptionReceived* represents the summed value of Energy, Gas, or Water
10706 received from the premises since the HFT on Monday to the last HFT read. If optionally provided,
10707 *CurrentWeekCostConsumptionReceived* is updated continuously as new measurements
10708 are made. If the optional *Historical Freeze Time* attribute is not available, default to midnight
10709 local time.

10710 D.7.2.2.6.11 PreviousWeekNCostConsumptionDelivered Attribute

10711 *PreviousWeekNCostConsumptionDelivered* represents the summed value of Energy, Gas, or Water
10712 delivered to the premises within the previous week period starting at the HFT on the
10713 Monday to the Sunday. If the optional *Historical Freeze Time* attribute is not available, default to
10714 midnight local time.

10715 D.7.2.2.6.12 PreviousWeekNCostConsumptionReceived Attribute

10716 *PreviousWeekNCostConsumptionReceived* represents the summed value of Energy, Gas, or
10717 Water received from the premises within the previous week period starting at the HFT on the
10718 Monday to the Sunday. If the optional *Historical Freeze Time* attribute is not available, default to
10719 midnight local time.

10720 D.7.2.2.6.13 CurrentMonthCostConsumptionDelivered Attribute

10721 *CurrentMonthCostConsumptionDelivered* represents the summed value of Energy, Gas, or Water
10722 delivered to the premises since the HFT on the 1st of the month to the last HFT read. If optionally
10723 provided, *CurrentMonthCostConsumptionDelivered* is updated continuously as new
10724 measurements are made. If the optional *Historical Freeze Time* attribute is not available, default
10725 to midnight local time.

10726 D.7.2.2.6.14 CurrentMonthCostConsumptionReceived Attribute

10727 *CurrentMonthCostConsumptionReceived* represents the summed value of Energy, Gas, or Water
10728 received from the premises since the HFT on the 1st of the month to the last HFT read. If optionally
10729 provided, *CurrentMonthCostConsumptionReceived* is updated continuously as new
10730 measurements are made. If the optional *Historical Freeze Time* attribute is not available, default
10731 to midnight local time.

10732 D.7.2.2.6.15 PreviousMonthNCostConsumptionDelivered Attribute

10733 *PreviousMonthNCostConsumptionDelivered* represents the summed value of Energy, Gas, or Water
10734 delivered to the premises within the previous Month period starting at the HFT on the 1st
10735 of the month to the last day of the month. If the optional *Historical Freeze Time* attribute is not
10736 available, default to midnight local time.

10737 D.7.2.2.6.16 PreviousMonthNCostConsumptionReceived Attribute

10738 *PreviousMonthNCostConsumptionReceived* represents the summed value of Energy, Gas, or
10739 Water received from the premises within the previous month period starting at the HFT on the 1st
10740 of the month to the last day of the month. If the optional *Historical Freeze Time* attribute is not
10741 available, default to midnight local time.

10742 D.7.2.2.6.17 HistoricalFreezeTime Attribute

10743 *HistoricalFreezeTime* represents the time of day, in Local Time, when Historical Cost
10744 Consumption attributes are captured. *HistoricalFreezeTime* is an unsigned 16-bit value
10745 representing the hour and minutes for HFT. The byte usages are:

10746 **Bits 0 to 7:** Range of 0 to 0x3B representing the number of minutes past the top of the hour.

10747 **Bits 8 to 15:** Range of 0 to 0x17 representing the hour of the day (in 24-hour format).

10749

10750

10751 D.7.2.3 Commands Received

10752 Table D-144 lists cluster-specific commands that are received by the server.

10753

10754

Table D-144— 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

10755

10756 D.7.2.3.1 Select Available Emergency Credit Command10757 This command is sent to the Metering Device to activate the use of any Emergency Credit available on the Metering Device.
1075810759 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)

Figure D-114— Select Available Emergency Credit Command Payload10760
10761

10762 D.7.2.3.1.2 Payload Details

10763 **Command Issue Date/Time (mandatory):** A UTCTime field to indicate the date and time at
10764 which the selection command was issued.

10765 **Originating Device (mandatory):** An 8-bit enumeration field identifying the SE device issuing
10766 the selection command, using the lower byte of the Device ID defined in Table 5-14,
10767 and summarized in Table D-145.

10768 **Table D-145– Originating Device Field Enumerations**

Enumerated Value	Device
0x00	Energy Service Interface
0x01	Meter
0x02	In-Home Display Device
0x03 – 0xFF	Reserved

10769

10770

10771 D.7.2.3.1.3 Effect on Receipt

10772 A Mirroring device receiving this command shall return a ZCL Default Response with a status
10773 code of NOTIFICATION_PENDING. If the command buffer on the mirror is already full, the
10774 mirror shall instead return a ZCL Default Response to the initiating device with a status code of
10775 INSUFFICIENT_SPACE.

10776

10777 D.7.2.3.2 Change Supply Command

10778 The *Change Supply* command has been moved to the Metering cluster (see Annex D.3 for further
10779 details).

10780 D.7.2.3.3 Change Debt Command

10781 The *ChangeDebt* command is sent to the Metering Device to change the debt values.

10782 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)

10783

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-115– Change Debt Command Payload

10784

D.7.2.3.3.2 Payload Details

10785 **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.

10786 **DebtLabel (mandatory):** The format and use of this field is the same as for the *DebtLabel#N* attribute as defined in D.7.2.2.3.1. A value of 0xFF in the first Octet (length) shall indicate that the value of this parameter shall remain unchanged on the Metering device following receipt of this command.

10787 **DebtAmount (mandatory):** The format and use of this field is the same as for the *DebtAmount#N* attribute as defined in D.7.2.2.3.2. A *DebtAmount* of 0xFFFFFFFF shall indicate that the value of this parameter shall remain unchanged on the Metering device following receipt of this command.

10788 **DebtRecoveryMethod (mandatory):** The format and use of this field is the same as for the *DebtRecoveryMethod#N* attribute as defined in D.7.2.2.3.3. A *DebtRecoveryMethod* of 0xFF shall indicate that the value of this parameter shall remain unchanged on the Metering device following receipt of this command.

10789 **DebtAmountType (mandatory):** An 8-bit enumeration field identifying the type of debt information to be issued within this command. The Types are detailed in Table D-146 below:

10800

Table D-146– 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

- 10807
10808 **DebtRecoveryStartTime (mandatory):** The format and use of this field is the same as for the
10809 *DebtRecoveryStartTime#N* attribute as defined in D.7.2.2.3.4. A *DebtRecoveryStartTime* of
10810 0xFFFFFFFF shall indicate that the value of this parameter shall remain unchanged on the
10811 Metering device following receipt of this command.
- 10812 **DebtRecoveryCollectionTime (mandatory):** The format and use of this field is the same as for
10813 the *DebtRecoveryCollectionTime#N* attribute as defined in D.7.2.2.3.5. A *DebtRecoveryCollectionTime* of
10814 0xFFFF shall indicate that the value of this parameter shall remain unchanged on the
10815 Metering device following receipt of this command.
- 10816 **DebtRecoveryFrequency (mandatory):** The format and use of this field is the same as for the
10817 *DebtRecoveryFrequency#N* attribute as defined in D.7.2.2.3.6. A *DebtRecoveryFrequency* of
10818 0xFF shall indicate that the value of this parameter shall remain unchanged on the Metering
10819 device following receipt of this command.
- 10820 **DebtRecoveryAmount (mandatory):** The format and use of this field is the same as for the
10821 *DebtRecoveryAmount#N* attribute as defined in D.7.2.2.3.7. A *DebtRecoveryAmount* of
10822 0xFFFFFFFF shall indicate that the value of this parameter shall remain unchanged on the
10823 Metering device following receipt of this command.
- 10824 **DebtRecoveryBalancePercentage (mandatory):** The format and use of this field is the same as
10825 for the *DebtRecoveryTopUpPercentage#N* attribute as defined in D.7.2.2.3.8. A *DebtRecoveryBalancePercentage* of
10826 0xFFFF shall indicate that the value of this parameter shall remain unchanged on the
10827 Metering device following receipt of this command.
- 10828 D.7.2.3.3.3 When Generated
10829 This command is generated when there is a change to the debt, which the Head End System
10830 requires to be sent down to the meter.
- 10831
- 10832 D.7.2.3.4 **Emergency Credit Setup Command**
10833 This command provides a method to set up the parameters for the Emergency Credit.
- 10834 D.7.2.3.4.1 Payload Format
- | Octets | 4 | 4 | 4 | 4 |
|-------------------|-------------------------|----------------|----------------------------|--------------------------------|
| Data Type | Unsigned 32-bit Integer | UTC Time | Unsigned 32-bit Integer | Unsigned 32-bit Integer |
| Field Name | Issuer Event ID (M) | Start Time (M) | Emergency Credit Limit (M) | Emergency Credit Threshold (M) |

Figure D-116– Emergency Credit Setup Command Payload

10836 D.7.2.3.4.2 Payload Details

10837 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. When
 10838 new information is provided that replaces older information for the same time period, this field
 10839 allows devices to determine which information is newer. The value contained in this field is a
 10840 unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)
 10841 identifying when the command was issued. Thus, newer information will have a value in the
 10842 *Issuer Event ID* field that is larger than older information.

10843 **Start Time (mandatory):** A UTC Time field to denote the time at which the Emergency Credit
 10844 settings become valid. A start date/time of 0x00000000 shall indicate that the command should
 10845 be executed immediately. A start date/time of 0xFFFFFFFF shall cause an existing but pending
 10846 *Emergency Credit Setup* command with the same *Issuer Event ID* to be cancelled.

10847 **Emergency Credit Limit (allowance) (mandatory):** An unsigned 32-bit field to denote the
 10848 Emergency Credit limit on the Metering Device, measured in base unit of *Currency* (as per the
 10849 Price cluster) or in Units (as per the Metering cluster) with the decimal point located as indicated
 10850 by the *TrailingDigits* field, as defined in the Price cluster. When no Emergency Credit has been
 10851 used, this is the value defined within the *EmergencyCreditRemaining* attribute (D.7.2.2.1.3).

10852 **Emergency Credit Threshold (mandatory):** An unsigned 32-bit field to denote the amount of
 10853 credit remaining on the Metering Device below which the Emergency Credit facility can be
 10854 selected. The value is measured in base unit of *Currency* (as per the Price cluster) or in Units (as
 10855 per the Metering cluster) with the decimal point located as indicated by the *TrailingDigits* field,
 10856 as defined in the Price cluster.

10857 D.7.2.3.4.3 When Generated

10858 The *Emergency Credit Setup* command is used when the Head End System has a requirement to
 10859 change the Prepayment configuration on the meter.

10860

10861 D.7.2.3.5 Consumer Top Up Command

10862 The *Consumer Top Up* command is used by the IHD and the ESI as a method to apply credit top
 10863 up values to a prepayment meter.

10864 D.7.2.3.5.1 Payload Format

Octets	1	1-25
Data Type	8 bit Enumeration	Octet String
Field Name	Originating Device (M)	TopUp Code (M)

10865 Figure D-117– Consumer Top Up Command Payload

10866 D.7.2.3.5.2 Payload Details

10867 **Originating Device (mandatory):** An 8 bit enumeration field identifying the Smart Energy
10868 device issuing the selection command, as defined in Table D-145.

10869 **Top Up Code (mandatory):** An octet string of between 1 and 25 characters (the first character
10870 indicates the string length).

10871 D.7.2.3.5.3 When Generated

10872 The *Consumer Top Up* command shall be generated when a new Top-up amount of credit has
10873 been purchased from the energy supplier and is required to be sent to the Meter. Alternatively,
10874 the command can be used to transfer an instruction such as to connect or disconnect the supply,
10875 enable a particular display sequence, or other action via an appropriate *Top Up (UTRN) Code*.

10876 D.7.2.3.5.4 Effect on Receipt

10877 The meter shall update the *Top Up Date/Time#1*, *Top Up Amount#1* and the *Originating
Device#1* attributes on the valid processing of this command. It shall then send the
10879 *ConsumerTopUpResponse* command to all devices bound to the cluster.

10880 A Mirroring device receiving this command shall return a ZCL Default Response with a status
10881 code of NOTIFICATION_PENDING. If the command buffer on the mirror is already full, the
10882 mirror shall instead return a ZCL Default Response to the initiating device with a status code of
10883 INSUFFICIENT_SPACE.

10884

10885 D.7.2.3.6 Credit Adjustment Command

10886 The *Credit Adjustment* command is sent to update the *Credit Remaining* attribute on a
10887 Prepayment meter. It shall only be sent from an ESI to the Meter.

10888 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)

10889 **Figure D-118– Credit Adjustment Command Payload**

10890 D.7.2.3.6.2 Payload Details

10891 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. When
10892 new information is provided that replaces older information for the same time period, this field

allows devices to determine which information is newer. The value contained in this field is a unique number managed by upstream servers or a UTC based time stamp (UTCTime data type) identifying when the command was issued. Thus, newer information will have a value in the *Issuer Event ID* field that is larger than older information.

Start Time (mandatory): A UTC Time field to denote the time at which the credit adjustment settings become valid. A start date/time of 0x00000000 shall indicate that the command should be executed immediately. A start date/time of 0xFFFFFFFF shall cause an existing but pending *Credit Adjustment* command with the same *Issuer Event ID* to be cancelled.

Credit Adjustment Type (mandatory): An 8-bit enumeration field identifying the type of credit adjustment to be issued out within this command. The Types are detailed within Table D-147 below.

Table D-147– Credit Type Field Enumerations

Enumerated Value	Credit Type
0x00	Credit Incremental
0x01	Credit Absolute
0x02 – 0xFF	Reserved

Credit Adjustment Value (mandatory): A signed 32-bit field to denote the value of the credit adjustment, measured in base unit of *Currency* (as per the Price cluster) or in Units (as per the Metering cluster) with the decimal point located as indicated by the *TrailingDigits* field, as defined in the Price cluster. This can be a positive or negative value.

10910 D.7.2.3.6.3 When Generated

10911 The *Credit Adjustment* command shall be sent to the meter when the ESI has a new credit
10912 adjustment value for the meter.

10913 D.7.2.3.6.4 Effect on Receipt

10914 The *Credit Adjustment Value* shall be used to update the *Credit Remaining* attribute to the correct
10915 value. If the value of the *Credit Adjustment Type* parameter is that of Emergency, then the
10916 amount of credit shall be added to the *EmergencyCreditRemaining* attribute; this shall not
10917 increase the *EmergencyCreditLimit* attribute.

10918

10919 D.7.2.3.7 Change Payment Mode Command

10920 This command is sent to a Metering Device to instruct it to change its mode of operation, e.g.
10921 from Credit to Prepayment.

10922 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)

10923 **Figure D-119– Change Payment Mode Command Payload**

10924 D.7.2.3.7.2 Payload Details

10925 **Provider ID (mandatory):** An unsigned 32 bit field containing a unique identifier for the
10926 commodity supplier to whom this command relates.

10927 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. When
10928 new information is provided that replaces older information for the same time period, this field
10929 allows devices to determine which information is newer. The value contained in this field is a
10930 unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)
10931 identifying when the command was issued. Thus, newer information will have a value in the
10932 *Issuer Event ID* field that is larger than older information.

10933 **Implementation Date/Time (mandatory):** A UTC Time field to indicate the date from which
10934 the payment mode change is to be applied. An *Implementation Date/Time* value of 0x00000000
10935 shall indicate that the command should be executed immediately. An *Implementation Date/Time*
10936 value of 0xFFFFFFFF shall cause an existing but pending *Change Payment Mode* command with
10937 the same *Provider ID* and *Issuer Event ID* to be cancelled.

10938 **Proposed Payment Control Configuration (mandatory):** An 16-bit BitMap indicating the
10939 actions required in relation to switching the payment mode. Bit encoding of this field is outlined
10940 in Table D-129.

10941 **Cut off Value (mandatory):** The format and use of this field is the same as for the *CutOffValue*
10942 attribute as defined in D.7.2.2.1.19. A *CutOffValue* of 0xFFFFFFFF shall indicate that the value
10943 of this parameter shall remain unchanged on the Metering device following receipt of this
10944 command.

10945 D.7.2.3.7.3 When Generated

10946 The *Change Payment Mode* command shall be sent from the Energy Supplier, via the ESI, only
10947 when the need to change the mode of the meter arises.

10948 D.7.2.3.7.4 Effect on Receipt

10949 On receipt of the *ChangePaymentMode* command, the meter shall send the
 10950 *ChangePaymentModeResponse*. The meter should create all snapshots required before the mode
 10951 is changed and transmit these to the ESI. It should then also create all required snapshots and
 10952 request valid Price, TOU and Prepayment information (refer to sections D.3.2.3.1.7 and D.7.2.4.2
 10953 for further details).

10954

10955 D.7.2.3.8 Get Prepay Snapshot Command

10956 This command is used to request the cluster server for snapshot data.

10957 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)

10958 **Figure D-120– Get Prepay Snapshot Command Payload**10959 D.7.2.3.8.2 Payload Details

10960 **Earliest Start Time (mandatory):** A UTC Timestamp indicating the earliest time of a snapshot
 10961 to be returned by a corresponding *Publish Prepay Snapshot* command. Snapshots with a time
 10962 stamp equal to or greater than the specified *Earliest Start Time* shall be returned.

10963 **Latest End Time (mandatory):** A UTC Timestamp indicating the latest time of a snapshot to be
 10964 returned by a corresponding *Publish Prepay Snapshot* command. Snapshots with a time stamp
 10965 less than the specified *Latest End Time* shall be returned.

10966 **Snapshot Offset (mandatory):** Where multiple snapshots satisfy the selection criteria specified
 10967 by the other fields in this command, this field identifies the individual snapshot to be returned.
 10968 An offset of zero (0x00) indicates that the first snapshot satisfying the selection criteria should be
 10969 returned, 0x01 the second, and so on.

10970 **Snapshot Cause (mandatory):** This field is used to request only snapshots for a specific cause.
 10971 The allowable values are listed in Table D-150. Setting the type to 0xFFFFFFFF indicates that all
 10972 snapshots should be transmitted, irrespective of the cause.

10973 D.7.2.3.8.3 Effect on Receipt

10974 On receipt of this command, the server will respond with the appropriate data as detailed in sub-
 10975 clause D.7.2.4.2.

10976 A ZCL Default Response with status NOT_FOUND shall be returned if the server does not have
 10977 a snapshot which satisfies the received parameters (e.g. no snapshot with a timestamp between
 10978 the *Earliest Start Time* and the *Latest End Time*).

10979

10980 D.7.2.3.9 **Get Top Up Log**

10981 This command is sent to the Metering Device to retrieve the log of Top Up codes received by the
 10982 meter.

10983 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)

10984 **Figure D-121– Get Top Up Code Log Command Payload**

10985 D.7.2.3.9.2 **Payload Details**

10986 **Latest End Time (mandatory):** UTC timestamp indicating the latest *TopUp Time* of Top
 10987 Up records to be returned by the corresponding *Publish Top Up Log* commands. The
 10988 first returned Top Up record shall be the most recent record with its *TopUp Time* equal
 10989 to or older than the *Latest End Time* provided.

10990 **Number of Records (mandatory):** An 8-bit integer which represents the maximum number of
 10991 records that the client is willing to receive in response to this command. A value of 0 would
 10992 indicate all available records shall be returned. The first returned Top Up record shall be the
 10993 most recent one in the log.

10994

10995 D.7.2.3.10 **Set Low Credit Warning Level**

10996 This command is sent from client to a Prepayment server to set the warning level for low credit.

10997 D.7.2.3.10.1 **Payload Format**

Octets	4
Data Type	Unsigned 32-bit Integer
Field Name	Low Credit Warning Level (M)

10998 **Figure D-122– Set Low Credit Warning Level Command Payload**

10999 D.7.2.3.10.2 **Payload Details**

11000 **Low Credit Warning Level (mandatory):** An unsigned 32 bit integer that defines the consumer
 11001 Low Credit value, in base unit of *Currency* (as per the Price cluster) or in Units (as per the

11002 Metering cluster), below which Low Credit warning should sound. The Low Credit warning shall
 11003 be triggered when the credit remaining on the meter falls below the value of the *Low Credit*
 11004 *Warning Level* above the disconnection point; this shall trigger the Low Credit Warning alert
 11005 within this cluster.

11006

11007 D.7.2.3.11 **Get Debt Repayment Log Command**

11008 This command is used to request the contents of the Repayment log.

11009 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

11010 **Figure D-123– GetDebtRepaymentLog Command Payload**

11011 D.7.2.3.11.2 **Payload Details**

11012 **Latest End Time (mandatory):** UTC timestamp indicating the latest *Collection Time* of
 11013 debt repayment records to be returned by the corresponding *Publish Debt Log* commands.
 11014 The first returned debt repayment record shall be the most recent record with its
 11015 *Collection Time* equal to or older than the *Latest End Time* provided.

11016 **Number of Debts (mandatory):** An 8-bit integer which represents the maximum number of debt
 11017 repayment records that the client is willing to receive in response to this command. A value of 0
 11018 would indicate all available records shall be returned. The first returned debt repayment record
 11019 shall be the most recent one in the log.

11020 **Debt Type (mandatory):** An 8-bit enumeration field identifying the type of debt record(s) to be
 11021 returned:

11022 **Table D-148– Debt Type Field Enumerations**

Enumerated Value	Debt Type
0x00	Debt 1
0x01	Debt 2
0x02	Debt 3
0x03 – 0xFE	Reserved
0xFF	All Debts

11023

11024

11025 D.7.2.3.12 **Set Maximum Credit Limit**

11026 This command is sent from a client to the Prepayment server to set the maximum credit level
 11027 allowed in the meter.

11028 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)

11029 **Figure D-124– Set Maximum Credit Level Command Payload**11030 D.7.2.3.12.2 Payload Details

11031 **Provider ID (mandatory):** An unsigned 32 bit field containing a unique identifier for the
 11032 commodity supplier to whom this command relates.

11033 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. When
 11034 new information is provided that replaces older information for the same time period, this field
 11035 allows devices to determine which information is newer. The value contained in this field is a
 11036 unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)
 11037 identifying when the command was issued. Thus, newer information will have a value in the
 11038 *Issuer Event ID* field that is larger than older information.

11039 **Implementation Date/Time (mandatory):** A UTC Time field to indicate the date from which
 11040 the maximum credit level is to be applied. An *Implementation Date/Time* of 0x00000000 shall
 11041 indicate that the command should be executed immediately. An *Implementation Date/Time* of
 11042 0xFFFFFFFF shall cause an existing but pending *Set Maximum Credit Limit* command with the
 11043 same *Provider ID* and *Issuer Event ID* to be cancelled.

11044 **Maximum Credit Level (mandatory):** An unsigned 32 bit integer value indicating the
 11045 maximum credit balance allowed on a meter. Any further top-up amount that will cause the
 11046 meter to exceed this limit will be rejected. This value can be stated in currency (as per the Price
 11047 cluster) or in units (unit of measure will be defined in the Metering cluster) depending on the
 11048 Prepayment mode of operation defined in section D.7.2.2.1.1 (*Payment Control Configuration*
 11049 attribute). A value of 0xFFFFFFFF will indicate that this limit is to be disabled and that all
 11050 further top-ups should be permitted.

11051 **MaximumCreditPerTopUp (mandatory):** An unsigned 32-bit integer value indicating the
 11052 maximum credit per top-up. Any single top-up greater than this threshold will cause the meter to
 11053 reject the top-up. This value can be stated in currency (as per the Price cluster) or in units (unit of
 11054 measure will be defined in the Metering cluster) depending on the Prepayment mode of operation
 11055 defined in section D.7.2.2.1.1 (*Payment Control Configuration* attribute). A value of
 11056 0xFFFFFFFF will indicate that this parameter is to be disabled and that there should be no limit
 11057 on the amount of credit allowed in a top-up.

11058

11059 D.7.2.3.13 **Set Overall Debt Cap**

11060 This command is sent from a client to the Prepayment server to set the overall debt cap allowed
 11061 in the meter.

11062 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

11063 **Figure D-125– Set Overall Debt cap Command Payload**11064 D.7.2.3.13.2 Payload Details

11065 **Provider ID (mandatory):** An unsigned 32 bit field containing a unique identifier for the
 11066 commodity supplier to whom this command relates.

11067 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. When
 11068 new information is provided that replaces older information for the same time period, this field
 11069 allows devices to determine which information is newer. The value contained in this field is a
 11070 unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)
 11071 identifying when the command was issued. Thus, newer information will have a value in the
 11072 *Issuer Event ID* field that is larger than older information.

11073 **Implementation Date/Time (mandatory):** A UTC Time field to indicate the date from which
 11074 the overall debt cap is to be applied. An *Implementation Date/Time* of 0x00000000 shall indicate
 11075 that the command should be executed immediately. An *Implementation Date/Time* of
 11076 0xFFFFFFFF shall cause an existing but pending *Set Overall Debt Cap* command with the same
 11077 *Provider ID* and *Issuer Event ID* to be cancelled.

11078 **Overall Debt Cap :** A signed 32 bit integer that defines the total amount of debt that can be
 11079 taken from top-ups (in the case of multiple instantiated top-up based debts on the Metering
 11080 Device) (see D.7.2.2.1.7). This field is always a monetary value, and as such the field is
 11081 measured in a base unit of *Currency* with the decimal point located as indicated by the *Trailing
 11082 Digits* field, as defined in the Price cluster.

11083

11084 D.7.2.4 Commands Generated

11085 Table D-149 lists commands that are generated by the server.

11086 **Table D-149– 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

11087

11088 D.7.2.4.1 **Supply Status Response Command**11089 The *Supply Status Response* command has been moved to the Metering cluster (see Annex D.3
11090 for further details).

11091

11092 D.7.2.4.2 **Publish Prepay Snapshot Command**11093 This command is generated in response to a *GetPrepaySnapshot* command. It is used to return a
11094 single snapshot to the client.11095 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)

11096 **Figure D-126– Publish Prepay Snapshot Command Payload**11097 D.7.2.4.2.2 **Payload Details**11098 **Snapshot ID (mandatory):** Unique identifier allocated by the device creating the snapshot.11099 **Snapshot Time (mandatory):** This is a 32 bit value (in UTC Time) representing the time at
11100 which the data snapshot was taken.11101 **Total Snapshots Found (mandatory):** An 8-bit Integer indicating the number of snapshots
11102 found, based on the search criteria defined in the associated *GetPrepaySnapshot* command. If the

11103 value is greater than 1, the client is able to request the next snapshot by incrementing the
 11104 *Snapshot Offset* field in an otherwise repeated *GetPrepaySnapshot* command.

11105 **Command Index (mandatory):** The *Command Index* is used to count the payload fragments in
 11106 the case where the entire payload does not fit into one message. The *Command Index* starts at 0
 11107 and is incremented for each fragment belonging to the same command.

11108 **Total Number of Commands (mandatory):** In the case where the entire payload does not fit
 11109 into one message, the *Total Number of Commands* field indicates the total number of sub-
 11110 commands in the message.

11111 **Snapshot Cause (mandatory):** A 32-bit BitMap indicating the cause of the snapshot. The
 11112 snapshot cause values are listed in Table D-150.

11113 **Table D-150—Snapshot Payload Cause**

Bit	Description
0	General
1	Reserved for Metering cluster
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

11114 NOTE: Where applicable, these Prepayment snapshots shall be taken in conjunction with the
 11115 associated snapshots in the Metering cluster.

11116 **SnapshotPayloadType (mandatory):** The *SnapshotPayloadType* is an 8-bit enumerator
 11117 defining the format of the *SnapshotPayload* in this message. The different snapshot types are
 11118 listed in Table D-151. The server selects the *SnapshotPayloadType* based on the charging
 11119 scheme in use.

11120 **Table D-151– Snapshot Payload Type**

Enumeration	Description
0x00	Debt/Credit Status
0x01 – 0xFE	Reserved
0xFF	Not used

11121
 11122 **SnapshotPayload (mandatory):** the format of the *SnapshotPayload* differs depending on the
 11123 *SnapshotPayloadType*.

11124 **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)

11125 **Figure D-127– Debt/Credit Status Command Payload**

11126 **Accumulated Debt (mandatory):** The *AccumulatedDebt* field represents the total amount of
 11127 debt remaining on the Metering Device, measured in a base unit of *Currency* with the decimal
 11128 point located as indicated by the *TrailingDigits* field, as defined in the Price cluster.

11129 **Type 1 Debt Remaining (mandatory):** The *Type1DebtRemaining* field represents the amount of
 11130 Type 1 debt remaining on the Metering Device, measured in base unit of *Currency* with the
 11131 decimal point located as indicated by the *TrailingDigits* field, as defined in the Price cluster.

11132 **Type 2 Debt Remaining (mandatory):** The *Type2DebtRemaining* field represents the amount of
 11133 Type 2 debt remaining on the Metering Device, measured in base unit of *Currency* with the
 11134 decimal point located as indicated by the *TrailingDigits* field, as defined in the Price cluster.

11135 **Type 3 Debt Remaining (mandatory):** The *Type3DebtRemaining* field represents the amount of
 11136 Type 3 debt remaining on the Metering Device, measured in base unit of *Currency* with the
 11137 decimal point located as indicated by the *TrailingDigits* field, as defined in the Price cluster.

11138 **Emergency Credit Remaining (mandatory):** The *EmergencyCreditRemaining* field represents
 11139 the amount of Emergency Credit still available on the Metering Device. If Monetary based, this
 11140 field is measured in a base unit of *Currency* (as per the Price cluster) or in Units (as per the
 11141 Metering cluster), with the decimal point located as indicated by the *TrailingDigits* field, as
 11142 defined in the Price cluster. If Unit based, the unit of measure is as defined in the Metering
 11143 cluster (see sub-clause D.3.2.2.4.1).

11144 **Credit Remaining (mandatory):** The *CreditRemaining* field represents the amount of credit
 11145 remaining on the Metering Device. If Monetary based, this field is measured in a base unit of
 11146 Currency (as per the Price cluster) or in Units (as per the Metering cluster), with the decimal
 11147 point located as indicated by the *TrailingDigits* field, as defined in the Price cluster. If Unit
 11148 based, the unit of measure is as defined in the Metering cluster (see sub-clause D.3.2.2.4.1).

11149

11150 D.7.2.4.3 Change Payment Mode Response Command

11151 This command is sent in response to the *ChangePaymentMode* command. The
 11152 *ChangePaymentModeResponse* command shall only inform the ESI of the current default setting
 11153 that would affect the meter when entering into Prepayment/PAYG or Credit mode. Should these
 11154 values require changing then other commands within the Prepayment & Price cluster should be
 11155 used.

11156 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)

11157 **Figure D-128– Change Payment Mode Response Command Payload**

11158 D.7.2.4.3.2 Payload Details

11159 **Friendly Credit (mandatory):** An 8-bit BitMap to show if the meter has a Friendly Credit
 11160 calendar and that this calendar shall be enabled.

11161 **Table D-152– Friendly Credit BitMap**

Bit	Description
0	Friendly credit enabled
1-7	Reserved

11162

11163 **Friendly Credit Calendar ID (mandatory):** An unsigned 32-bit field to denote the
 11164 IssuerCalendarID that shall be used for the friendly credit periods. The *IssuerCalendarID* can be
 11165 found within the TOU cluster (see Annex D.9).

11166 **Emergency Credit Limit/Allowance (mandatory):** An unsigned 32-bit field to denote the
 11167 emergency credit limit on the Metering Device, measured in base unit of *Currency* with the
 11168 decimal point located as indicated by the *TrailingDigits* field, as defined in the Price cluster.
 11169 Should no emergency credit have been used, this is the value defined within the
 11170 EmergencyCreditRemaining attribute (D.7.2.2.1.3).

11171 **Emergency Credit Threshold (mandatory):** An unsigned 32-bit field to denote the amount of
 11172 credit remaining on the Metering Device below which the *Emergency Credit* facility can be
 11173 selected. The value is measured in base unit of *Currency* with the decimal point located as
 11174 indicated by the *TrailingDigits* field, as defined in the Price cluster.

11175 D.7.2.4.3.3 When Generated

11176 The *ChangePaymentModeResponse* command is generated in response to a
 11177 *ChangePaymentMode* command.

11178

11179 D.7.2.4.4 Consumer Top Up Response Command

11180 The Metering device responds either with the following values in the case of a credit token
 11181 received:

- 11182 • Meter's enumerated status, after receiving the top up, in the *Result Type* field
- 11183 • Received Top up token's credit value in the *Top Up Value* field
- 11184 • The source of the top up, enumerated in the *Source of Top up* field
- 11185 • The credit remaining on the meter, after the addition of this Top Up, in the *Credit Remaining*
 11186 field,

11187 OR, in the case of a connect/disconnect Top Up (UTRN) code, with the following:

- 11188 • Supply status, after processing of the token, enumerated in the *Result Type* field
- 11189 • Top up token's credit value SET TO ZERO in the *Top Up Value* field
- 11190 • The source of the top up, enumerated in the *Source of Top up* field
- 11191 • The credit remaining on the meter, after the addition of this Top Up, in the *Credit Remaining*
 11192 field

11193 D.7.2.4.4.1 Payload Format

Octets	1	4	1	4
--------	---	---	---	---

Data Type	8-bit Enumeration	Signed 32-bit Integer	8-bit Enumeration	Signed 32-bit Integer
Field Name	Result Type (M)	Top Up Value (M)	Source of Top up (M)	Credit Remaining (M)

Figure D-129– Consumer Top Up Response Command Payload

11194

11195 D.7.2.4.4.2 Payload Details11196 **Result Type (mandatory):** An 8-bit enumerated value indicating whether the Metering Device
11197 accepted or rejected the top up. Enumerated values are described in Table D-153

11198

Table D-153– 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

11199

11200 **Top up Value (mandatory):** A signed 32-bit integer field representing the Top Up value
11201 available in the top up content. If it is Monetary based, this field is measured in a base unit of
11202 *Currency* with the decimal point located as indicated by the Trailing Digits field, as defined in
11203 the Price cluster. If Unit based, the unit of measure is as defined in the Metering cluster (see sub-
11204 clause D.3.2.2.4.1). If *Result Type* is other than *Accepted*, this field has a maximum value
11205 (0xFFFFFFFF) which indicates an invalid Top Up value.11206 **Source of Top Up (mandatory):** An 8-bit enumeration indicating the device that has issued the
11207 top up (see Table D-145 for applicable enumerations).11208 **Credit Remaining (mandatory):** The *Credit Remaining* field represents the amount of credit
11209 remaining on the Metering Device after addition of a top up. If Monetary based, this field is
11210 measured in a base unit of *Currency* with the decimal point located as indicated by the *Trailing*
11211 *Digits* field, as defined in the Price cluster. If Unit based, the unit of measure is as defined in the
11212 Metering cluster (see sub-clause D.3.2.2.4.1). In case of *Result Type* other than *Accepted*, the

11213 Credit Remaining field has a maximum value (0xFFFFFFFF) representing invalid credit
 11214 remaining.

11215 D.7.2.4.4.3 When Generated

11216 The *ConsumerTopUpResponse* command is generated in response to a *ConsumerTopUp*
 11217 command.

11218

11219

11220 D.7.2.4.5 Publish Top Up Log Command

11221 This command is used to send the Top Up Code Log entries to the client. They are sent most
 11222 recent entry first.

11223 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

11224 **Figure D-130– Publish Top Up Log Command Payload**

11225 D.7.2.4.5.2 Payload Details

11226 **Command Index (mandatory):** The *Command Index* is used to count the payload fragments in
 11227 the case where the entire payload does not fit into one message. The *Command Index* starts at 0
 11228 and is incremented for each fragment belonging to the same command. A value of 0xFE in the
 11229 *Command Index* states that this is the last expected command.

11230 **Total Number of Commands (mandatory):** In the case that an entire payload does not fit into
 11231 one message, the *Total Number of Commands* field indicates the total number of sub-commands
 11232 in the message.

11233 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)

11234 **Figure D-131– Top Up Code Payload**

11235 **TopUp Code (mandatory):** This is the value of the Top Up code stored in the log.

11236 **TopUp Amount (mandatory):** This is the amount of credit that was added to the Metering
 11237 Device during this Top Up.

11238 **TopUp Time (mandatory):** This is the UTC Timestamp when the Top Up was applied to the
 11239 Metering Device.

11240

11241 D.7.2.4.6 **Publish Debt Log Command**

11242 This command is used to send the contents of the Repayment Log.

11243 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)

11244 **Figure D-132– Publish Debt Log Command Payload**

11245 D.7.2.4.6.2 **Payload Details**

11246 **Command Index (mandatory):** The *Command Index* is used to count the payload fragments in
 11247 the case where the entire payload does not fit into one message. The *Command Index* starts at 0
 11248 and is incremented for each fragment belonging to the same command. A value of 0xFE in the
 11249 *Command Index* states that this is the last expected command.

11250 **Total Number of Commands (mandatory):** In the case that an entire payload does not fit into
 11251 one message, the *Total Number of Commands* field indicates the total number of sub-commands
 11252 in the message.

11253 **Debt Payload (mandatory):**

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)

11254 **Figure D-133– Format of the Debt Payload**

11255 **Collection Time (mandatory):** An UTC time field identifying the time when the collection
 11256 occurred.

11257 **Amount Collected (mandatory):** An unsigned 32-bit field to denote the amount of debt
 11258 collected at this time. This parameter shall be measured in base unit of *Currency* with the
 11259 decimal point located as indicated by the *Trailing Digits* field, as defined in the Price cluster.

11260 **Debt Type (mandatory):** An 8-bit enumeration field identifying the type of debt the record
 11261 refers to. The enumerations are defined in Table D-148.

11262 **Outstanding Debt (mandatory):** An unsigned 32-bit field to denote the amount of debt still
11263 outstanding after the debt was collected. This parameter shall be measured in base unit of
11264 Currency with the decimal point located as indicated by the *Trailing Digits* field, as defined in
11265 the Price cluster.

11266

11267 **D.7.3 Client**

11268 **D.7.3.1 Dependencies**

- 11269 • Support for ZCL Data Types
- 11270 • Events carried using this cluster include a timestamp with the assumption that target devices
11271 maintain a real time clock. Devices can acquire and synchronize their internal clocks via the
11272 ZCL Time server.

11273 **D.7.3.2 Attributes**

11274 The client has no attributes.

11275 **D.7.3.3 Commands Received**

11276 The client receives the cluster-specific response commands detailed in D.7.2.4.

11277 **D.7.3.4 Commands Generated**

11278 The client generates the cluster-specific commands detailed in D.7.2.3, as required by the
11279 application.

11280

11281 **D.7.4 Application Guidelines**

11282 **D.7.4.1 Credit Status Attribute**

11283 The purpose of the *Credit Status* attribute is to describe to any device on the HAN, what the
11284 status of a meter operating in Prepayment mode may be at any point in time. There are a number
11285 of important functionalities in Prepayment meters, and a variety of implementations depending
11286 on the manufacturer and their chosen system, however this attribute is designed to pick up the
11287 lowest common denominator of statuses that would be important to an end user looking to glean
11288 information about their meter in the HAN. For example, has their meter run out of credit, is
11289 Emergency Credit available or has Emergency Credit been selected?

11290 The diagram below describes the manner in which this attribute SHOULD be used when
11291 describing these statuses and others. This guidance note is not designed to prescribe how any
11292 Prepayment meter logic works, but merely to get a common understanding of the meter status to

11293 the end users' interface device. It is entirely up to device manufacturers to decide how to best use
 11294 this information and display it.

11295

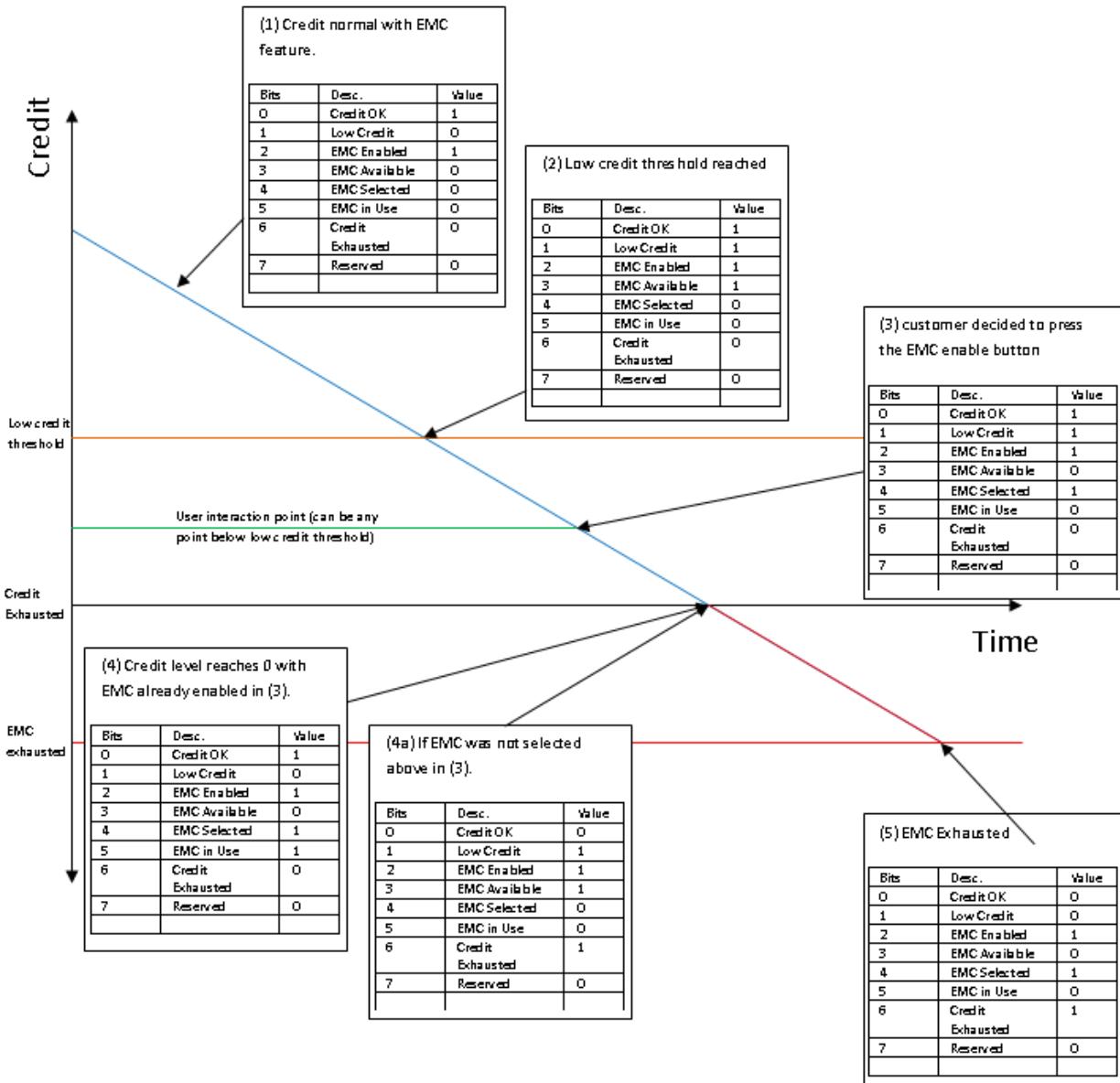


Figure D-134– Prepayment Credit Status Attribute Explained

11296
 11297

11298

11299 D.7.4.1.1 **Statuses Explained - an Example**

11300 Below is a brief explanation of each status noted on the diagram above in order to give a better
11301 indication of what the meter is doing at any given point. Imagine that the diagonal blue line
11302 represents the customer's credit, and when it turns into a plum colored diagonal line below the
11303 Time-Axis, the reader can assume that the meter is in negative credit, and Emergency Credit may
11304 or may not be invoked depending on the use case.

11305 The definitions of functionality below are modeled on the current understanding of Prepayment
11306 functionality. However there could well be a situation when meters are not disconnected when
11307 reaching the zero credit point, or indeed when Emergency Credit has been exhausted. This
11308 description is designed to aid understanding only and not specify meter functionality (see Figure
11309 D-134):

11310 1. At this stage the meter has customer credit and has the Emergency Credit feature enabled.
11311 This means that when the meter reaches the Low Credit threshold, Emergency Credit will be
11312 available to be selected by the end user.

11313 2. At this point the meter still has customer credit available, but the meter has now reached the
11314 Low Credit threshold. This means that the end user may, should they choose to do so, select to
11315 engage the Emergency Credit. This will allow the meter to pass into a predefined amount of
11316 negative credit, without disconnection, when the meter credit reaches zero. The Emergency
11317 Credit can be selected at any point below the Low Credit threshold, but if this is not done
11318 before the customer's credit reaches zero then the meter will disconnect the supply.

11319 3. Same as above except this is demonstrating the point at which the end user actually engages
11320 the Emergency Credit function, and in doing so making Emergency Credit no longer available
11321 for selection again.

11322 4. Meter reaches zero credit with Emergency Credit function engaged. This means that the
11323 option to engage Emergency Credit functionality is not available to the end user (as he has
11324 already done it), but the meter is still connected and 'Credit OK' remains set because
11325 Emergency Credit is available.

11326 a. In this case the end user has decided not to engage Emergency Credit functionality
11327 before the credit level reaches zero, thereby removing the 'Credit OK' flag once the
11328 available credit has reached zero. The Emergency Credit function is still available,
11329 but requires end user interaction in order to engage it.

11330 5. At this point Emergency Credit is exhausted and the meter is assumed to have disconnected
11331 (this may not be the case depending on the supplier's requirements). There is no available
11332 credit or Emergency Credit, and it is not possible for the end user to engage the Emergency
11333 Credit function.

11334 At this point in time, when all credit is exhausted, the meter and IHD will need to display the
11335 "debt to clear". This is the amount of credit that must be put onto the meter in order to exceed

11336 the Low Credit warning threshold and get the meter back on supply, with Emergency Credit
11337 available again (credit above zero will get the lights back on but Emergency Credit will not be
11338 available until credit is above the Low Credit Warning Threshold). The ‘debt to clear’ will be
11339 transmitted by way of the Credit Remaining register (as it will be a negative number at this
11340 time, made up of the debt that the meter has accrued while in Emergency Credit).

11341 If Standing Charge, debt repayment charges and energy charges are normally being paid, these
11342 may not all be charged during an Emergency Credit period, but will still accrue in the
11343 background until Emergency Credit is exhausted (at point 5). Depending on energy supplier
11344 preference, it SHALL be configurable whether or not Emergency Credit is used to pay debt
11345 charges. The Emergency Credit value, along with debt charges accrued in the background
11346 while Emergency Credit was in operation, will be added to the ‘debt to clear’ register in the
11347 meter when Emergency Credit is exhausted, and displayed on the Credit Remaining register as
11348 a negative number.

11349

11350

11351 **D.8 Over-the-Air Bootload Cluster**11352 **D.8.1 Overview**

11353 The over-the-air bootload cluster provides a common mechanism to manage and serve up
11354 upgrade images for devices from different manufacturers in the same network. Servers provide
11355 firmware images to clients to download, controlling the timing for downloads and when the
11356 actual upgrade to a new version of software is made. Clients periodically query the server for
11357 new images and then can download the image at a rate according to their capabilities or
11358 policies.

11359 Details for the over-the-air (OTA) bootload cluster are maintained in a separate document,
11360 reference [095264r15].

11361 Smart Energy devices may optionally support the over-the-air bootload cluster client or server.
11362 If the OTA cluster is implemented by a Smart Energy device then APS encryption on all unicast
11363 messages shall be used. Smart Energy devices that implement the client must support ECDSA
11364 signature verification of images.

11365 Additionally, over-the-air bootload cluster client devices that are intended to be field
11366 upgradeable to Smart Energy 2.0 should support the optional feature “query specific file” in
11367 order to potentially receive device specific data necessary for the transition to a Smart Energy 2.0
11368 device.

11369 **D.8.2 OTA Bootloading Timing Considerations**

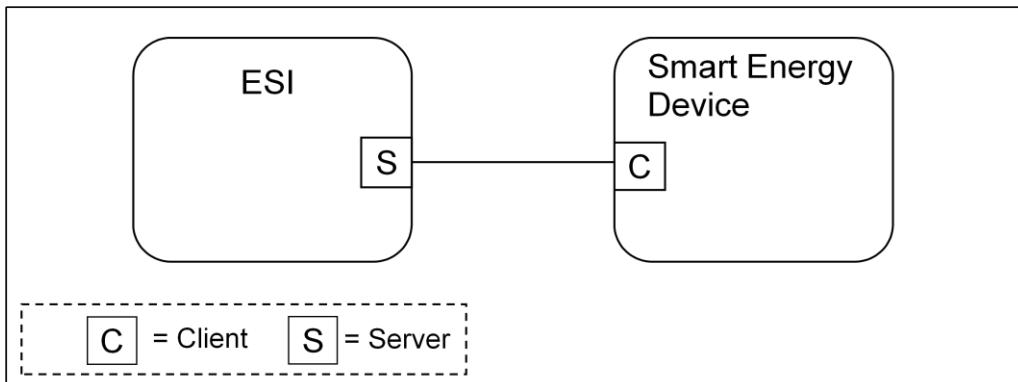
11370 The OTA cluster defines the message formatting used to pass device images but does not
11371 specify when to use the cluster. The following policies specify how and when to use the OTA
11372 cluster such that all devices in an SE network will upgrade at predictable intervals.

- 11373 1 OTA clients shall perform service discovery to find the OTA server after registration has
11374 completed.
- 11375 2 An OTA client device that does not find an OTA server in the network shall periodically
11376 attempt a new discovery once a day.
- 11377 3 All devices shall query the OTA server at least once a day for information about the next
11378 version to upgrade to. Non-sleepy devices in the network may be instructed to begin a new
11379 download at any point in time via the *Image Notify* command.
- 11380 4 All client devices may download data as quickly as their capabilities allow, but at a minimum
11381 rate of one block per 10 minutes. This means that at a rate of 1 block (50 bytes) per 10
11382 minutes, a 128k file will take 18 days to download.

11383 **D.9 Calendar Cluster**11384 **D.9.1 Overview**

11385 The Calendar cluster implements commands to transfer calendar information within the
 11386 premises. The calendar information is distributed by an ESI.

11387



11388
11389

Figure D-135—Calendar Cluster

11390 The server shall be able to store at least **two** instances of the calendar, typically the current and
 11391 the next one. It is recommended that a client is also capable of storing 2 instances. It is also
 11392 recommended that a Calendar server may additionally store at least **one** previous instance of the
 11393 calendar.

11394 The Calendar server shall send unsolicited *PublishCalendar* and *PublishSpecialDays* commands
 11395 to its clients if they are bound to it. Other calendar items such as Day Profiles, Week Profiles and
 11396 Season information shall not be sent unsolicited. The clients shall send corresponding Get...
 11397 commands to fetch the information from the server as necessary. The Calendar server shall
 11398 publish new calendars, to clients that have bound to receive them, as soon as they become
 11399 available. Devices with limited resources, and which cannot therefore handle multiple calendars,
 11400 should NOT ‘register’ (i.e. bind to the server) to receive unsolicited Calendar cluster commands.
 11401 If there is no next calendar available, a ZCL Default Response shall be returned with status
 11402 NOT_FOUND; the ESI shall publish the information as soon as it gets it from the HES. Devices
 11403 (particularly battery-powered devices) should regularly check for updates to calendar items.

11404 The Calendar must be replaced as a whole; only the Special Day Table can be changed
 11405 independently. To uniquely identify the parts of a calendar, an Issuer Calendar ID is used. All
 11406 parts belonging to the same calendar must have the same Issuer Calendar ID. All parts of a
 11407 particular calendar shall be successfully retrieved from the server before a client can use that
 11408 calendar. It is anticipated that a change to any part of a calendar, other than a Special Day Table,
 11409 will result in a new calendar and a new Issuer Calendar ID.

11410 The Calendar cluster will support all of the following calendar types:

11411 • Delivered

11412 • Received

11413 • Delivered and Received

11414 • Friendly Credit

11415 • Auxiliary Load Switch

11416 Each calendar has three associated tables, a Season table, a Week Profile table and a Day Profile table. These are described in Table D-154. In addition, there is a Special Day Table which allows
11417 special days to be defined (days where a special switching behavior overrides the normal
11418 operation). Each entry in the Special Day table contains a date together with the Day ID for a
11419 Day Profile (in the associated Calendar's Day Profile table) to be used on that date.
11420

11421

Table D-154—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>

11422

11423

11424 All dates and times shall be defined according to UTC. The Season Table may be used to
11425 accommodate requirements such as daylight saving. Alternatively, calendars can be defined
11426 according to Standard or Local time.

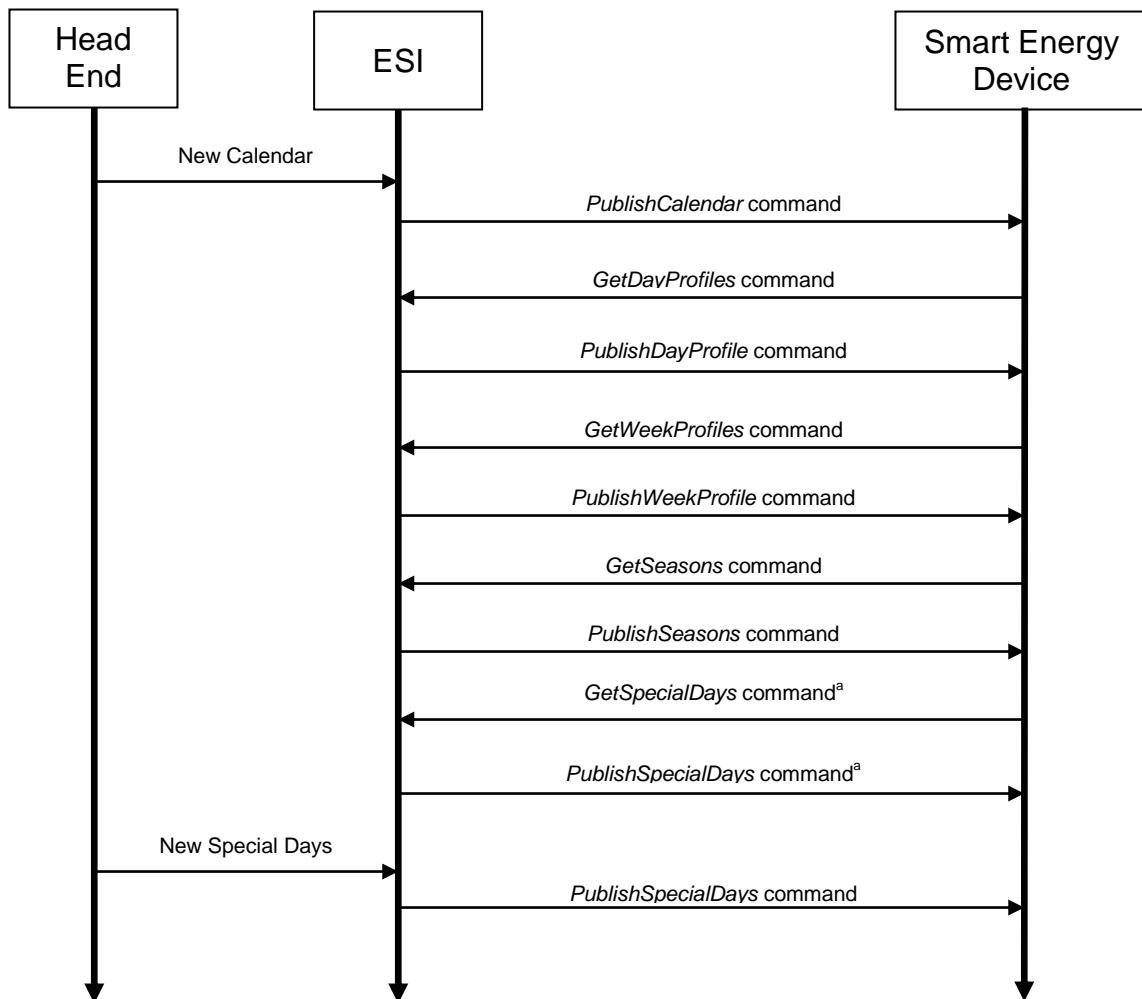
11427
11428 Figure D-136 shows a recommended Calendar command sequence (noting that this sequence is
11429 for a main-powered Smart Energy Device) :

11430

11431

11432

11433



11461 ^aAlthough not necessary, it is thought wise to check for updates when a new calendar is published
11462
11463

Figure D-136– Recommended Calendar Command Sequence

11464

11465 **D.9.2 Server**11466 **D.9.2.1 Dependencies**

11467 A device implementing the Calendar server shall also implement the Price server. A device
 11468 implementing the Calendar client shall also implement the Price client. The commodity type of a
 11469 Calendar server shall be inferred from that of the corresponding Price server (i.e. located on the
 11470 same device/endpoint). It is expected that the TOU calendar and tariff information of the Price
 11471 cluster is provided by the same utility supplier. The *ProviderID* for the TOU calendar shall be
 11472 obtained from the *Tariff Information Set* of the Price Cluster.

11473 **D.9.2.2 Attributes**

11474 For convenience, the attributes defined in this cluster are arranged into sets of related
 11475 attributes; each set can contain up to 256 attributes. Attribute identifiers are encoded such that
 11476 the most significant Octet specifies the attribute set and the least significant Octet specifies the
 11477 attribute within the set. The currently defined attribute sets are listed in the following Table
 11478 D-155.

11479 **Table D-155—Calendar Cluster Attribute Sets**

Attribute Set Identifier	Description
0x00	Auxiliary Switch Label Attribute Set
0x01 – 0xFF	Reserved

11480

11481 **D.9.2.2.1 Auxiliary Switch Label Attribute Set**11482 **Table D-156—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					
--------------------	----------	--	--	--	--	--

11483

11484 D.9.2.2.1.1 AuxSwitchNLabel Attributes

11485 The *AuxSwitchNLabel* attributes provide a method for assigning a label to an Auxiliary Switch.
 11486 The *AuxSwitchNLabel* attributes are ZCL Octet String fields capable of storing 22-character
 11487 strings (the first Octet indicates length) encoded in the UTF-8 format.

11488

11489 D.9.2.3 Commands Generated

11490 Table D-157 lists commands that are generated by the server.

11491

Table D-157– 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

11492

11493 D.9.2.3.1 PublishCalendar Command

11494 The *PublishCalendar* command is published in response to a *GetCalendar* command or if new
 11495 calendar information is available. The Calendar must be replaced as a whole; only the Special
 11496 Day Table can be changed independently. All parts of a calendar instance shall have the same
 11497 Calendar ID.

11498 Nested and overlapping calendars are not allowed. In the case of overlapping calendars of the
 11499 same type (calendar type), the calendar with the newer *IssuerCalendarID* takes priority over all
 11500 nested and overlapping calendars. All existing calendar instances that overlap, even partially,
 11501 should be removed. The only exception to this is if a calendar instance with a newer *Issuer Event*
 11502 *ID* overlaps with the end of the current active calendar but is not yet active, then the active
 11503 calendar is not deleted but modified so that the active calendar ends when the new calendar
 11504 begins.

11505 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)

11506

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)

11507

Figure D-137 – PublishCalendar Command Payload

11508

D.9.2.3.1.2 Payload Details

11509 **Provider Id (mandatory):** An unsigned 32-bit field containing a unique identifier for the
 11510 commodity provider. This field allows differentiation in deregulated markets where multiple
 11511 commodity providers may be available.

11512 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. When
 11513 new information is provided that replaces older information for the same time period, this field
 11514 allows devices to determine which information is newer. The value contained in this field is a
 11515 unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)
 11516 identifying when the Publish command was issued. Thus, newer information will have a value in
 11517 the *Issuer Event ID* field that is larger than older information.

11518 **Issuer Calendar ID (mandatory):** Unique identifier generated by the commodity Supplier to
 11519 identify a particular calendar.

11520 **Start Time (mandatory):** A UTC Time field to denote the time at which the published calendar
 11521 becomes valid. A start date/time of 0x00000000 shall indicate that the command should be
 11522 executed immediately.

11523 **Calendar Type (mandatory):** An 8-bit enumeration identifying the type of calendar published
 11524 in this command. Table D-158 details the enumeration of this field. Generation Meters shall use
 11525 the ‘Received’ Calendar.

11526

Table D-158 – Calendar Type Enumeration

Value	Description
0x00	Delivered Calendar
0x01	Received Calendar

Value	Description
0x02	Delivered and Received Calendar
0x03	Friendly Credit Calendar
0x04	Auxillary Load Switch Calendar
0x05-0xFF	Reserved

11527

11528 **Calendar Time Reference (mandatory):** This field indicates how the switching times contained
 11529 in the calendar are to be interpreted. The following table shows possible values:

11530

Table D-159– Calendar Time Reference Enumeration

Value	Description
0x00	UTC time
0x01	Standard time
0x02	Local time
0x03-0xFF	Reserved

11531

11532 Standard time refers to UTC time adjusted according to the local time zone.

11533 Local time refers to Standard time adjusted according to local daylight savings regulations.

11534 Where the optional Standard and/or Local Time (as applicable) are not available on the Time
 11535 cluster server, the *Calendar Time Reference* shall default to UTC time.

11536 **Calendar Name (mandatory):** The *CalendarName* provides a method for utilities to assign a
 11537 name to the entire calendar. The *CalendarName* is a ZCL Octet String field capable of storing a
 11538 12 character string (the first Octet indicates length) encoded in the UTF-8 format.

11539 **Number of Seasons (mandatory):** Number of entries in the Seasons Table. A value of 0x00
 11540 means no Season defined.

11541 **Number of Week Profiles (mandatory):** Number of week profiles in the Week Profile Table. A
 11542 value of 0x00 means no Week Profile defined.

11543 **Number of Day Profiles (mandatory):** Number of day profiles in the Day Profile Table.
 11544

11545 D.9.2.3.2 PublishDayProfile Command

11546 The *PublishDayProfile* command is published in response to a *GetDayProfile* command. If the
 11547 *IssuerCalendarID* does not match with one of the stored calendar instances, the client shall
 11548 ignore the command and respond using ZCL Default Response with a status response of
 11549 NOT_FOUND.

11550 The Calendar server shall send only the number of Schedule Entries belonging to this calendar instance. Server and clients shall be able to store at least 1 *DayProfile* for TOU and Auxiliary Load Switch calendars and three *DayProfiles* for a Friendly Credit calendar, and at least one *ScheduleEntries* per day profile. If the client is not able to store all *ScheduleEntries*, the device should respond using ZCL Default Response with a status response of INSUFFICIENT_SPACE.

11555 The ESI may send as many *PublishDayProfile* commands as needed, if the maximum application payload is not sufficient to transfer all *ScheduleEntries* in one command. In this case:

- 11557 • The *ScheduleEntries* shall be arranged in a linear array ordered by the start time.
- 11558 • The first command shall have *CommandIndex* set to 0, the second to 1 and so on.
- 11559 • The *Total Number of Commands* sub-field shall be set in all commands to the total number of **commands** being transferred.
- 11562 • The *Total Number of Schedule Entries* field shall be set in all commands to the total number of **entries** being transferred with the whole set of commands.
- 11563 • All associated commands shall use the same value of *Issuer Event ID*.

11564 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)

11565

Octets	1	Variable
Data Type	8-bit Enumeration	Series of Schedule Entries
Field Name	Calendar Type (M)	Day Schedule Entries

11566 **Figure D-138– *PublishDayProfile* Command Payload**

11567 D.9.2.3.2.2 Payload Details

11568 **Provider Id (mandatory):** An unsigned 32-bit field containing a unique identifier for the
11569 commodity provider. This field allows differentiation in deregulated markets where multiple
11570 commodity providers may be available.

11571 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. When
11572 new information is provided that replaces older information for the same time period, this field
11573 allows devices to determine which information is newer. The value contained in this field is a
11574 unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)
11575 identifying when the Publish command was issued. Thus, newer information will have a value in
11576 the Issuer Event ID field that is larger than older information.

11577 **Issuer Calendar ID (mandatory):** Unique identifier generated by the commodity supplier. All
11578 parts of a calendar instance shall have the same *Issuer Calendar ID*.

11579 **Day ID (mandatory):** Unique identifier generated by the commodity supplier. The *Day ID* is
11580 used as reference to assign a Day Profile to a Special Day or days in a Week Profile. When
11581 generating calendars, *Day IDs* shall be allocated sequentially, starting from 1.

11582 **Total Number of Schedule Entries (mandatory):** An 8-bit integer representing the total
11583 number of *ScheduleEntries* in this Day Profile.

11584 **Command Index (mandatory):** The *CommandIndex* is used to count the payload fragments in
11585 the case where the entire payload does not fit into one message. The *CommandIndex* starts at 0
11586 and is incremented for each fragment belonging to the same command.

11587 **Total Number of Commands (mandatory):** In the case where the entire payload does not fit
11588 into one message, the *Total Number of Commands* field indicates the total number of sub-
11589 commands in the message.

11590 **Calendar Type (mandatory):** An 8-bit enumeration identifying the type of calendar published
11591 in this command. Table D-158 details the enumeration of this field. This field identifies the type
11592 of *Day Schedule Entry* included in this command.

11593 D.9.2.3.2.3 Day Schedule Entries

11594 The format of Day Schedule entries is dependent on the Calendar Type (see Table D-158). If the
11595 Calendar Type is 0x00 – 0x02 then Rate Switch Times shall be used. If the value is 0x03 then the
11596 FriendlyCreditSwitch times shall be used. If the value is 0x04 then the Auxilliary Load Switch
11597 times shall be used. A value other than these would be invalid.

11598 **D.9.2.3.2.3.1 Schedule Entries for Rate Switch Times**

11599 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)

11600

Figure D-139– Schedule Entries for Rate Switch Times Command Sub-Payload

11601 **Start Time (mandatory):** The *Start Time* is represented in minutes from midnight.
 11602 *ScheduleEntries* must be arranged in ascending order of *Start Times*. The first Schedule Entry
 11603 must have 0x0000 (midnight) as the StartTime.

11604 **Price Tier (mandatory):** This is the current price tier that is valid until the start time of the next
 11605 Schedule Entry.

11606 **D.9.2.3.2.3.2 Schedule Entries for Friendly Credit Switch Times**

11607 A *Friendly Credit Switch Time* entry consists of a start time and an indication if Friendly Credit
 11608 is available.

Octets	2	1
Data Type	Unsigned 16-bit Integer	Boolean
Field Name	Start Time (M)	Friendly Credit Enable (M)

Figure D-140– Schedule Entries for Friendly Credit Switch Times Command Sub-Payload

11610 **Start Time (mandatory):** The *Start Time* is represented in minutes from midnight.
 11611 *ScheduleEntries* must be arranged in ascending order of *Start Times*. The first Schedule Entry
 11612 must have 0x0000 (midnight) as the StartTime.

11613 **Friendly Credit Enable (mandatory):** The *Friendly Credit Enable* field is a Boolean denoting
 11614 if the Friendly Credit period is available for the consumer to use. A value of 1 means it is enabled
 11615 and a 0 means that the Friendly Credit period is not available for the consumer to use.

11616 **D.9.2.3.2.3.3 Schedule Entries for Auxilliary Load Switch Times**

11617 An *Auxilliary Load Switch Time* entry consists of a start time, the relevant Auxiliary Switch and
 11618 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)

11619 **Figure D-141– Schedule Entries for Auxilliary Load Switch Times Command Sub-Payload**

11620 **Start Time (mandatory):** The *Start Time* is represented in minutes from midnight.
 11621 *ScheduleEntries* must be arranged in ascending order of *Start Times*. The first Schedule Entry
 11622 must have 0x0000 (midnight) as the StartTime.

11623 **Auxiliary Load Switch State (mandatory):** The required status of the auxiliary switches is
 11624 indicated by the state of the bits. Bit0 correspond to Auxiliary Switch 1 and bit7 corresponds to
 11625 Auxiliary Switch 8. A bit set to “1” indicates an ON state and a bit set to “0” indicates an OFF
 11626 state.

11627 D.9.2.3.3 **PublishWeekProfile Command**

11628 The *PublishWeekProfile* command is published in response to a *GetWeekProfile* command. If the
 11629 *IssuerCalendarID* does not match with one of the stored calendar instances, the client shall
 11630 ignore the command and respond using ZCL Default Response with a status response of
 11631 NOT_FOUND.

11632 The Calendar server shall send only the number of WeekProfiles belonging to this calendar
 11633 instance. Server and clients shall be able to store at least 4 WeekProfiles for TOU calendars, and
 11634 1 WeekProfile for Friendly Credit and Auxiliary Load Switch calendars. If the client is not able
 11635 to store all entries, the device should respond using ZCL Default Response with a status response
 11636 of INSUFFICIENT_SPACE.

11637 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)

11638

1	1	1	1	1	1	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

11639 **Figure D-142– PublishWeekProfile Command Payload**

11640 D.9.2.3.3.2 **Payload Details**

11641 **Provider Id (mandatory):** An unsigned 32-bit field containing a unique identifier for the
 11642 commodity provider. This field allows differentiation in deregulated markets where multiple
 11643 commodity providers may be available.

11644 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. When
 11645 new information is provided that replaces older information for the same time period, this field
 11646 allows devices to determine which information is newer. The value contained in this field is a
 11647 unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)
 11648 identifying when the Publish command was issued. Thus, newer information will have a value in
 11649 the *Issuer Event ID* field that is larger than older information.

11650 **Issuer Calendar ID (mandatory):** Unique identifier generated by the commodity supplier. All
 11651 parts of a calendar instance shall have the same *Issuer Calendar ID*.

11652 **Week ID (mandatory):** Unique identifier generated by the commodity supplier. The *Week ID* is
 11653 used as reference to assign a Week Profile to a Season Entry. When generating calendars, *Week*
 11654 *IDs* shall be allocated sequentially, starting from 1.

11655 **Day ID Ref Monday until Day ID Ref Sunday (mandatory):** Reference to the related Day
 11656 Profile entry.

11657 D.9.2.3.4 PublishSeasons Command

11658 The *PublishSeasons* command is published in response to a *GetSeason* command. If the
 11659 *IssuerCalendarID* does not match with one of the stored calendar instances, the client shall
 11660 ignore the command and respond using ZCL Default Response with a status response of
 11661 NOT_FOUND.

11662 The Calendar server shall send only the number of *SeasonEntries* belonging to this calendar
 11663 instance. Server and clients shall be able to store at least 4 *SeasonEntries* for TOU calendars, and
 11664 1 *SeasonEntry* for Friendly Credit and Auxiliary Load Switch calendars. If the client is not able
 11665 to store all *Season Entries*, the device should respond using ZCL Default Response with a status
 11666 response of INSUFFICIENT_SPACE.

11667 The ESI may send as many *PublishSeasons* commands as needed, if the maximum application
 11668 payload is not sufficient to transfer all Season Entries in one command. In this case:

- 11669 • The *SeasonEntries* shall be arranged in a linear array ordered by the date.
- 11670 • The first command shall have *Command Index* set to 0, the second to 1 and so on.
- 11671 • The total number of seasons being transferred with the whole set of commands is known
 11672 from the previously received *PublishCalendar* command.
- 11673 • All associated commands shall use the same value of *Issuer Event ID*.

11674 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

11675

Figure D-143– PublishSeasons Command Payload11676 D.9.2.3.4.2 Payload Details

Provider Id (mandatory): An unsigned 32-bit field containing a unique identifier for the commodity provider. This field allows differentiation in deregulated markets where multiple commodity providers may be available.

Issuer Event ID (mandatory): Unique identifier generated by the commodity provider. When new information is provided that replaces older information for the same time period, this field allows devices to determine which information is newer. The value contained in this field is a unique number managed by upstream servers or a UTC based time stamp (UTCTime data type) identifying when the Publish command was issued. Thus, newer information will have a value in the *Issuer Event ID* field that is larger than older information.

Issuer Calendar ID (mandatory): Unique identifier generated by the commodity supplier. All parts of a calendar instance shall have the same *Issuer Calendar ID*.

Command Index (mandatory): The *Command Index* is used to count the payload fragments in the case where the entire payload does not fit into one message. The *Command Index* starts at 0 and is incremented for each fragment belonging to the same command.

Total Number of Commands (mandatory): In the case where the entire payload does not fit into one message, the *Total Number of Commands* field indicates the total number of sub-commands in the message.

Season Entry: A *Season Entry* consists of a *Season Start Date* and the reference (*Week ID Ref*) to the related Week Profile entry. The Start Date of the *Season Entries* must be arranged in 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)

11697

Figure D-144– Season Entry Sub-Payload11698 D.9.2.3.5 PublishSpecialDays Command

The *PublishSpecialDays* command is published in response to a *GetSpecialDays* command or if a calendar update is available. If the *Calendar Type* does not match with one of the stored

11701 calendar instances, the client shall ignore the command and respond using ZCL Default
 11702 Response with a status response of NOT_FOUND.

11703 The Calendar server shall send only the number of *SpecialDayEntries* belonging to this calendar
 11704 instance. Server and clients shall be able to store at least 50 *SpecialDayEntries*. If the client is not
 11705 able to store all *SpecialDayEntries*, the device should respond using ZCL Default Response with
 11706 a status response of INSUFFICIENT_SPACE.

11707 If the maximum application payload is not sufficient to transfer all *SpecialDayEntries* in one
 11708 command, the ESI may send as many *PublishSpecialDays* commands as needed. In this case:

- 11709 • The *SpecialDayEntries* shall be arranged in a linear array ordered by the date.
- 11710 • The first command shall have *Command Index* set to 0, the second to 1 and so on.
- 11711 • The *Total Number of SpecialDays* field shall be set in all commands to the total number of
 11712 entries being transferred with the whole set of commands.
- 11713 • All associated commands shall use the same value of *Issuer Event ID*.

11714 Note that, in this case, it is the client's responsibility to ensure that it receives all associated
 11715 *PublishSpecialDays* commands before any of the payloads can be used.

11716 D.9.2.3.5.1 Payload Format

11717 The *PublishSpecialDays* command shall be formatted as illustrated in Figure D-145:

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)

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

11719

Figure D-145– *PublishSpecialDays* Command Payload

11720 D.9.2.3.5.2 Payload Details

11721 **Provider Id (mandatory):** An unsigned 32-bit field containing a unique identifier for the
11722 commodity provider. This field allows differentiation in deregulated markets where multiple
11723 commodity providers may be available.

11724 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. When
11725 new information is provided that replaces older information for the same time period, this field
11726 allows devices to determine which information is newer. The value contained in this field is a
11727 unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)
11728 identifying when the Publish command was issued. Thus, newer information will have a value in
11729 the Issuer Event ID field that is larger than older information. If multiple *PublishSpecialDays*
11730 commands are needed to transfer the whole Special Day Table, the commands belonging to the
11731 same Special Day Table shall use the same *IssuerEventID* and *StartTime*.

11732 **Issuer Calendar ID (mandatory):** Unique identifier generated by the commodity Supplier. All
11733 parts of a calendar instance shall have the same *Issuer Calendar ID*.

11734 **Start Time (mandatory):** A UTC Time field to denote the time at which the Special Day Table
11735 becomes valid. A start date/time of 0x00000000 shall indicate that the command should be
11736 executed immediately. A start date/time of 0xFFFFFFFF shall cause an existing
11737 *PublishSpecialDays* command with the same *Provider ID* and *Issuer Event ID* to be cancelled
11738 (note that, in markets where permanently active price information is required for billing
11739 purposes, it is recommended that a replacement/superseding *PublishSpecialDays* command is
11740 used in place of this cancellation mechanism).

11741 **Calendar Type (mandatory):** An 8-bit enumeration identifying the type of calendar this day
11742 profile belongs to. Generation Meters shall use the ‘Received’ Calendar. See Table D-158.

11743 **Total Number of SpecialDays (mandatory):** An 8-bit integer representing the total number of
11744 Special Day entries in this Special Day Table.

11745 **Command Index (mandatory):** The *Command Index* is used to count the payload fragments in
11746 the case where the entire payload does not fit into one message. The *Command Index* starts at 0
11747 and is incremented for each fragment belonging to the same command.

11748 **Total Number of Commands (mandatory):** In the case where the entire payload does not fit
11749 into one message, the *Total Number of Commands* field indicates the total number of sub-
11750 commands in the message.

11751 **SpecialDayEntry:** A *SpecialDayEntry* consists of the *Special Day Date* and a reference (*Day ID*
11752 *Ref*) to the related Day Profile entry. The dates of the Special Day Table must be arranged in
11753 ascending order.

Octets	4	1
--------	---	---

Octets	4	1
Data Type	Date	Unsigned 8-bit Integer
Field Name	Special Day Date (M)	Day ID Ref (M)

11754

Figure D-146– *SpecialDayEntry* Sub-Payload

11755

11756 D.9.2.3.6 **CancelCalendar Command**11757 The *CancelCalendar* command indicates that all data associated with a particular calendar
11758 instance should be discarded.11759 In markets where permanently active price (and hence calendar) information is required for
11760 billing purposes, it is recommended that replacement/superseding *PublishCalendar*,
11761 *PublishDayProfile*, *PublishWeekProfile* and *PublishSeasons* commands are used in place of a
11762 *CancelCalendar* command. The exception is a ‘Friendly Credit’ calendar, where an instance is
11763 not always required.11764 D.9.2.3.6.1 **Payload Format**11765 The *CancelCalendar* command shall be formatted as illustrated in Figure D-147:

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)

11766

Figure D-147– *CancelCalendar* Command Payload11767 D.9.2.3.6.2 **Payload Details**11768 **Provider Id (mandatory):** An unsigned 32-bit field containing a unique identifier for the
11769 commodity provider. This field allows differentiation in deregulated markets where multiple
11770 commodity providers may be available.11771 **Issuer Calendar ID (mandatory):** Unique identifier generated by the commodity Supplier. All
11772 parts of a calendar instance shall have the same *Issuer Calendar ID*.11773 **Calendar Type (mandatory):** An 8-bit enumeration identifying the type of calendar to be
11774 cancelled by this command. Table D-158 details the enumeration of this field.

11775 D.9.2.3.6.3 Effect on Receipt

11776 On receipt of this command, a client device shall discard all instances of *PublishCalendar*,
 11777 *PublishDayProfile*, *PublishWeekProfile*, *PublishSeasons* and *PublishSpecialDays* commands
 11778 associated with the stated *Provider ID*, *Calendar Type* and *Issuer Calendar ID*.

11779

11780 D.9.2.4 Commands Received

11781 Table D-160 lists cluster-specific commands that are received by the server.

11782 **Table D-160– 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

11783

11784 D.9.2.4.1 GetCalendar Command

11785 This command initiates *PublishCalendar* command(s) for scheduled Calendar updates. To obtain
 11786 the complete Calendar details, further *GetDayProfiles*, *GetWeekProfiles* and *GetSeasons*
 11787 commands must be sent using the *IssuerCalendarID* obtained from the appropriate
 11788 *PublishCalendar* command.

11789 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)

11790 **Figure D-148– GetCalendar Command Payload**

11791 D.9.2.4.1.2 Payload Details

11792 **Earliest Start Time (mandatory):** UTC Timestamp indicating the earliest start time of
 11793 calendars to be returned by the corresponding *PublishCalendar* command. The first returned
 11794 *PublishCalendar* command shall be the instance which is active or becomes active at or after the

11795 stated *Earliest Start Time*. If more than one instance is requested, the active and scheduled
 11796 instances shall be sent with ascending ordered *Start Time*.

11797 **Min. Issuer Event ID (mandatory):** A 32-bit integer representing the minimum *Issuer Event ID*
 11798 of calendars to be returned by the corresponding *PublishCalendar* command. A value of
 11799 0xFFFFFFFF means not specified; the server shall return calendars irrespective of the value of
 11800 the *Issuer Event ID*.

11801 **Number of Calendars (mandatory):** An 8-bit integer which represents the maximum number of
 11802 *PublishCalendar* commands that the client is willing to receive in response to this command. A
 11803 value of 0 would indicate all available *PublishCalendar* commands shall be returned.

11804 **Calendar Type (mandatory):** An 8-bit enumeration identifying the calendar type of the
 11805 requested calendar. Generation Meters shall use the ‘Received’ Calendar. See Table D-158. A
 11806 value of 0xFF means not specified. If the *CalendarType* is not specified, the server shall return
 11807 calendars regardless of its type.

11808 **Provider Id (mandatory):** An unsigned 32-bit field containing a unique identifier for the
 11809 commodity provider. This field allows differentiation in deregulated markets where multiple
 11810 commodity providers may be available. A value of 0xFFFFFFFF means not specified; the server
 11811 shall return calendars irrespective of the value of the *Provider Id*.

11812 **D.9.2.4.2 GetDayProfiles Command**

11813 This command initiates one or more *PublishDayProfile* commands for the referenced Calendar.

11814 **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)

11815 **Figure D-149– GetDayProfiles Command Payload**

11816 **D.9.2.4.2.2 Payload Details**

11817 **Provider Id (mandatory):** An unsigned 32-bit field containing a unique identifier for the
 11818 commodity provider. This field allows differentiation in deregulated markets where multiple
 11819 commodity providers may be available. A value of 0xFFFFFFFF means not specified; the server
 11820 shall return day profiles irrespective of the value of the *Provider Id*.

11821 **Issuer Calendar ID (mandatory):** *IssuerCalendarID* of the calendar to which the requested
 11822 Day Profiles belong.

11823 **Start Day ID (mandatory):** Unique identifier for a Day Profile generated by the commodity
 11824 supplier. The *Start Day ID* indicates the minimum ID of Day Profiles to be returned by the
 11825 corresponding *PublishDayProfile* command. A value of 0x01 indicates that the (first)
 11826 *PublishDayProfile* command should contain the profile with the lowest Day ID held by the
 11827 server. A value of 0x00 is unused.

11828 **Number of Days (mandatory):** An 8-bit integer which represents the maximum number of Day
 11829 Profiles that the client is willing to receive in response to this command. A value of 0x00 will
 11830 cause the return of all day profiles with an ID equal to or greater than the *Start Day ID*.

11831 Note: A Day Profile table may need multiple *PublishDayProfile* commands to be transmitted to
 11832 the client.

11833 **D.9.2.4.3 GetWeekProfiles Command**

11834 This command initiates one or more *PublishWeekProfile* commands for the referenced Calendar.

11835 **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)

11837 **Figure D-150– GetWeekProfiles Command Payload**

11838 **D.9.2.4.3.2 Payload Details**

11839 **Provider Id (mandatory):** An unsigned 32-bit field containing a unique identifier for the
 11840 commodity provider. This field allows differentiation in deregulated markets where multiple
 11841 commodity providers may be available. A value of 0xFFFFFFFF means not specified; the server
 11842 shall return week profiles irrespective of the value of the *Provider Id*.

11843 **Issuer Calendar ID (mandatory):** *IssuerCalendarID* of the calendar to which the requested
 11844 Week Profiles belong.

11845 **Start Week ID (mandatory):** Unique identifier for a Week Profile generated by the commodity
 11846 supplier. The *Start Week ID* indicates the minimum ID of Week Profiles to be returned by the
 11847 corresponding *PublishWeekProfile* command. A value of 0x01 indicates that the
 11848 *PublishWeekProfile* command should contain the profile with the lowest Week ID held by the
 11849 server. A value of 0x00 is unused.

11850 **Number of Weeks (mandatory):** An 8-bit integer which represents the maximum number of
 11851 Week Profiles that the client is willing to receive in response to this command. A value of 0x00
 11852 will cause the return of all week profiles with an ID equal to or greater than the *Start Week ID*.

11853 D.9.2.4.4 **GetSeasons Command**

11854 This command initiates one or more *PublishSeasons* commands for the referenced Calendar.

11855 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)

11856 **Figure D-151– GetSeasons Command Payload**

11857 **Provider Id (mandatory):** An unsigned 32-bit field containing a unique identifier for the
 11858 commodity provider. This field allows differentiation in deregulated markets where multiple
 11859 commodity providers may be available. A value of 0xFFFFFFFF means not specified; the server
 11860 shall return season tables irrespective of the value of the *Provider Id*.

11861 **Issuer Calendar ID (mandatory):** *IssuerCalendarID* of the calendar to which the requested
 11862 Seasons belong.

11863 Note: A Season Table may need multiple *PublishSeasons* commands to be transmitted to the
 11864 client.

11865 D.9.2.4.5 **GetSpecialDays Command**

11866 This command initiates one or more *PublishSpecialDays* commands for the scheduled Special
 11867 Day Table updates.

11868 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)

11869 **Figure D-152– GetSpecialDays Command Payload**11870 D.9.2.4.5.2 **Payload Details**

11871 **Start Time (mandatory):** UTC Timestamp to select active and scheduled events to be returned
 11872 by the corresponding *PublishSpecialDays* command. If the command has a *Start Time* of
 11873 0x00000000, replace that *Start Time* with the current time stamp.

11874 **Number of Events (mandatory):** An 8-bit integer which represents the maximum number of
 11875 Special Day Table instances to be sent. A value of 0 would indicate all available Special Day
 11876 tables shall be returned. The first returned *PublishSpecialDays* command should be that which is
 11877 active or becomes active at the stated *Start Time*. The first returned Special Day table shall be the

11878 instance which is active or becomes active at the stated *Start Time*. If more than one instance is
11879 requested, the active and scheduled instances shall be sent with ascending ordered *Start Time*.

11880 Note: A Special Day table may need multiple *PublishSpecialDay* commands to be transmitted to
11881 the client.

11882 **Calendar Type (mandatory):** An 8-bit enumeration identifying the calendar type of the
11883 requested Special Days. Generation Meters shall use the ‘Received’ Calendar. See Table D-158.
11884 A value of 0xFF means not specified. If the *CalendarType* is not specified, the server shall return
11885 Special Days regardless of their type.

11886 **Provider Id (mandatory):** An unsigned 32-bit field containing a unique identifier for the
11887 commodity provider. This field allows differentiation in deregulated markets where multiple
11888 commodity providers may be available. A value of 0xFFFFFFFF means not specified; the server
11889 shall return Special Day tables irrespective of the value of the *Provider Id*.

11890 **Issuer Calendar ID (mandatory):** Unique identifier generated by the commodity supplier. A
11891 value of 0x00000000 will cause the return of all Special Days profiles.

11892

11893 D.9.2.4.6 [GetCalendarCancellation Command](#)

11894 This command initiates the return of the last *CancelCalendar* command held on the associated
11895 server.

11896 D.9.2.4.6.1 [Payload Details](#)

11897 This command has no payload.

11898 D.9.2.4.6.2 [When Generated](#)

11899 This command is generated when the client device wishes to fetch any pending *CancelCalendar*
11900 command from the server (see D.9.2.3.6 for further details). In the case of a BOMD, this may be
11901 as a result of the associated Notification flag.

11902 A ZCL Default response with status NOT_FOUND shall be returned if there is no
11903 *CancelCalendar* command available.

11904

11905 D.9.3 Client

11906 D.9.3.1 Dependencies

11907 Support for ZCL Data Types.

11908 No dependencies exist for other Smart Energy clusters.

11909 [**D.9.3.2 Attributes**](#)

11910 The client has no attributes.

11911 [**D.9.3.3 Commands Received**](#)

11912 The client receives the cluster-specific response commands detailed in D.9.2.3.

11913 [**D.9.3.4 Commands Generated**](#)

11914 The client generates the cluster-specific commands detailed in D.9.2.4, as required by the
11915 application.

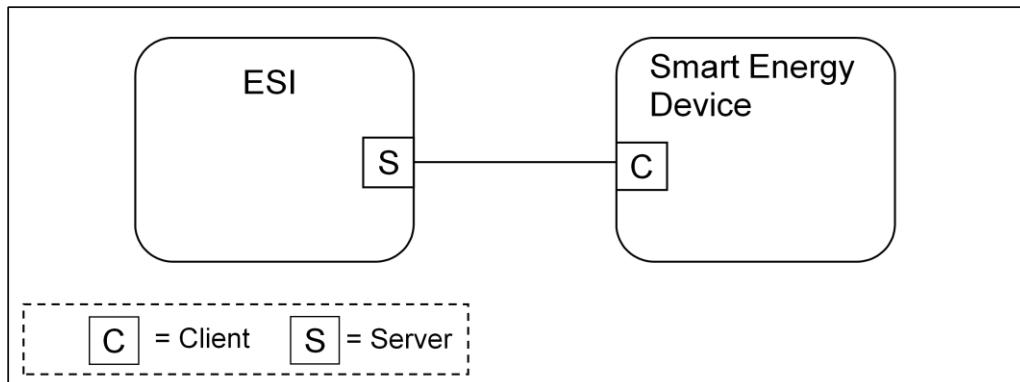
11916

11917 **D.10 Device Management Cluster**

11918 **D.10.1 Overview**

11919 The Device Management Cluster provides an interface to the functionality of devices within a
11920 Smart energy Network. The cluster will support the following functions:

- 11921 • Supplier Control
11922 • Tenancy Control
11923 • Password Control
11924 • Event Configuration



11925
11926 **Figure D-153– Device Management Cluster Client/Server Example**

- 11927
- 11928 **D.10.1.1 Supplier Control**
11929 This functionality provides a method to control the activities required to change the energy
11930 supplier to the premises (CoS).
- 11931 **D.10.1.2 Tenancy Control**
11932 This functionality provides a method to control the activities required when changing the tenant
11933 (consumer) of the property (CoT).
- 11934 **D.10.1.3 Password Control**
11935 Passwords or PINs are used to protect access to consumer data or to secure access to the energy
11936 supplier's meter service menus.
- 11937 The Password commands provide a mechanism where a specific password located on a Smart
11938 Energy device may be changed to a new value or reset. The server shall maintain an access
11939 control list of the type of password required vs. the device and, where applicable, store the last

11940 password for the device. Each device that supports this feature shall have a local default
 11941 password.

11942 The server shall send unsolicited *RequestNewPasswordResponse* commands to its clients (except
 11943 BOMDs unless unsolicited messages are enabled in its policy) when the backhaul connection
 11944 requires the device to update the password.
 11945

11946 D.10.2 Server

11947 D.10.2.1 Dependencies

11948 Events carried using this cluster include a timestamp with the assumption that target devices
 11949 maintain a real time clock. Devices can acquire and synchronize their internal clocks via the ZCL
 11950 Time server.

11951

11952 D.10.2.2 Attributes

11953 For convenience, the attributes defined in this specification are arranged into sets of related
 11954 attributes; each set can contain up to 256 attributes. Attribute identifiers are encoded such that the
 11955 most significant Octet specifies the attribute set and the least significant Octet specifies the
 11956 attribute within the set. The currently defined attribute sets are listed in the following Table
 11957 D-161.

11958 **Table D-161– 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

11959

11960 D.10.2.2.1 Supplier Control Attribute Set

11961 **Table D-162– 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 – 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 – 0x01FF	Reserved					

11962

11963 D.10.2.2.1.1 Provider ID Attribute11964 An unsigned 32-bit field containing a unique identifier for the current commodity supplier. The
11965 default value of 0x00000000 shall be used for installation.

- 11966 D.10.2.2.1.2 Provider Name Attribute
 11967 An octet string containing the name of the current supplier of the commodity to the device. The
 11968 attribute is capable of storing a 16 character string (the first octet indicates length) encoded in the
 11969 UTF-8 format.
- 11970 D.10.2.2.1.3 Provider Contact Details Attribute
 11971 An octet string containing the contact details of the current Provider delivering a commodity to
 11972 the premises. The attribute is capable of storing a 19 character string (the first octet indicates
 11973 length) encoded in UTF-8 format.
- 11974 D.10.2.2.1.4 Proposed Provider ID Attribute
 11975 An unsigned 32-bit field containing a unique identifier for the commodity supplier associated
 11976 with the proposed change to the supply of the commodity.
- 11977 D.10.2.2.1.5 Proposed Provider Name Attribute
 11978 The *Proposed Provider Name* indicates the name for the commodity supplier associated with the
 11979 proposed change to the supply of energy. This attribute is an octet string field capable of storing
 11980 a 16 character string (the first octet indicates length) encoded in the UTF-8 format.
- 11981 D.10.2.2.1.6 Proposed Provider Change Date/Time Attribute
 11982 A UTC time that defines the time and date when the new supplier will take over the supply of the
 11983 commodity to the Meter/HAN.
- 11984 D.10.2.2.1.7 Proposed Provider Change Control
 11985 This is a 32-bit mask that denotes the functions that are required to be carried out on processing
 11986 of the change of supplier. The format of this Bitmap is shown within Table D-163.

Table D-163– 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-164. 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	

11988
11989**Table D-164– Contactor State Bit Combinations**

Bit Combination	Status
0b00	Supply OFF
0b01	Supply OFF / ARMED
0b10	Supply ON (see note)
0b11	Supply UNCHANGED

11990

- 11991 D.10.2.2.1.8 ReceivedProviderID Attribute
 11992 An unsigned 32-bit field containing a unique identifier for the commodity supplier receiving the
 11993 Received energy.
- 11994 D.10.2.2.1.9 ReceivedProviderName Attribute
 11995 The name of the current supplier of Received energy services to the device. This attribute is an
 11996 octet string field capable of storing a 16 character string (the first octet indicates length) encoded
 11997 in the UTF-8 format.
- 11998 D.10.2.2.1.10 ReceivedProviderContactDetails Attribute
 11999 An octet string containing the contact details of the current Provider receiving a commodity from
 12000 the premises. The attribute is capable of storing a 19 character string (the first octet indicates
 12001 length) encoded in UTF-8 format.

12002 D.10.2.2.1.11 ReceivedProposedProviderID Attribute

12003 An unsigned 32-bit field containing a unique identifier for the commodity supplier associated
12004 with the proposed change to the Receiving of energy.

12005 D.10.2.2.1.12 ReceivedProposedProviderName Attribute

12006 The *Received Proposed Provider Name* indicates the name for the commodity supplier
12007 associated with the proposed change to the Receiving of energy. This attribute is an octet string
12008 field capable of storing a 16 character string (the first octet indicates length) encoded in the UTF-
12009 8 format.

12010 D.10.2.2.1.13 ReceivedProposedProviderChangeDate/Time Attribute

12011 A UTC time that defines the time and date that the new supplier will take over the Received of
12012 energy from the Meter/HAN.

12013 D.10.2.2.1.14 ReceivedProposedProviderChangeControl Attribute

12014 This is a 32-bit mask that denotes the functions that are required to be carried out on processing
12015 of the change of supplier. The format of this Bitmap is shown within Table D-163.

12016 D.10.2.2.2 Tenancy Control Attribute Set

12017 **Table D-165 – 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					

12018

12019 D.10.2.2.2.1 ChangeofTenancyUpdateDate/Time Attribute

12020 The *ChangeofTenancyUpdateDate/Time* attribute indicates the time at which a proposed change
12021 to the tenancy is to be implemented. Until an initial change of tenancy becomes available, this
12022 attribute shall be set to 0xFFFFFFFF (i.e. invalid).

12023 D.10.2.2.2.2 ProposedTenancyChangeControl Attribute

12024 This is a 32-bit mask that denotes the functions that are required to be carried out on processing
12025 of the change of tenancy. The format of this Bitmap is shown within Table D-163. Until an initial
12026 change of tenancy becomes available, this attribute shall be set to 0x00000000.

12027 D.10.2.2.3 **Backhaul Control Attribute Set**

12028 **Table D-166– 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					

12029

12030 D.10.2.2.3.1 **WAN Status Attribute**

12031 The *WAN Status* attribute is an 8-bit enumeration defining the state of the WAN (Wide Area Network) connection as listed in the table below:

12033 **Table D-167– State of the WAN Connection**

Enumeration	Description
0x00	Connection to WAN is not available
0x01	Connection to WAN is available
0x02 - 0xFF	Reserved

12034

12035 D.10.2.2.4 **HAN Control Attribute Set**

12036 **Table D-168– HAN Control Attribute Set**

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					

12037

12038 D.10.2.2.4.1 **Low Medium Threshold Attribute**

12039 The *Low Medium Threshold* attribute is an unsigned 32-bit integer indicating the threshold at which the value of *Instantaneous Demand* is deemed to have moved from low energy usage to medium usage. The unit of measure for this value is as specified by the *UnitOfMeasure* attribute within the Metering cluster (see Table D-26 for definition).

12043 D.10.2.2.4.2 Medium High Threshold Attribute

12044 The *Medium High Threshold* attribute is an unsigned 32-bit integer indicating the threshold at
 12045 which the value of *Instantaneous Demand* is deemed to have moved from medium energy usage
 12046 to high usage. The unit of measure for this value is as specified by the *UnitOfMeasure* attribute
 12047 within the Metering cluster (see Table D-26 for definition).

12048

12049 D.10.2.3 Commands Received

12050 Table D-169 lists the cluster specific commands that are received by the server.

12051

Table D-169– 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	

12052

12053 D.10.2.3.1 Get Change of Tenancy Command

12054 This command is used to request the ESI to respond with information regarding any available
 12055 change of tenancy.

12056 D.10.2.3.1.1 Payload Details

12057 There are no fields for this command.

12058 D.10.2.3.1.2 Effect on Receipt

12059 The ESI shall send a *PublishChangeofTenancy* command. A ZCL Default Response with status
 12060 NOT_FOUND shall be returned if there is no change of tenancy information available.

12061

12062 D.10.2.3.2 Get Change of Supplier Command

12063 This command is used to request the ESI to respond with information regarding any available
 12064 change of supplier.

12065 D.10.2.3.2.1 Payload Details

12066 There are no fields for this command.

12067 D.10.2.3.2.2 Effect on Receipt

12068 The ESI shall send a *PublishChangeofSupplier* command. A ZCL Default Response with status
12069 NOT_FOUND shall be returned if there is no change of supplier information available.

12070

12071 D.10.2.3.3 RequestNewPassword Command

12072 This command is used to request the current Password from the server.

12073 D.10.2.3.3.1 Payload Format

Octets	1
Data Type	Unsigned 8-bit Enumeration
Field Name	Password Type (M)

12074 **Figure D-154– RequestNewPassword Command Payload**

12075 D.10.2.3.3.2 Payload Details

12076 **PasswordType (mandatory):** Indicates which password is requested. The possible password
12077 types are defined in Table D-171.

12078

12079 D.10.2.3.4 GetSiteID Command

12080 This command is used to request the ESI to respond with information regarding any pending
12081 change of Site ID.

12082 D.10.2.3.4.1 Payload Details

12083 There are no fields for this command.

12084 D.10.2.3.4.2 Effect on Receipt

12085 The ESI shall send an *UpdateSiteID* command. A ZCL Default Response with status
12086 NOT_FOUND shall be returned if there is no change of Site ID pending.

12087

12088 D.10.2.3.5 Report Event Configuration Command

12089 This command is sent in response to a *GetEventConfiguration* command.

12090 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)

12091 **Figure D-155– Report Event Configuration Command Payload**

12092 D.10.2.3.5.2 Payload Details

12093 **Command Index (mandatory):** The *Command Index* is used to count the payload fragments in
12094 the case where the entire payload does not fit into one message. The *Command Index* starts at 0
12095 and is incremented for each fragment belonging to the same command.

12096 **Total Commands (mandatory):** This parameter holds the total number of responses.

12097 **Event Configuration Payload (mandatory):** The log payload is a series of events, in time
12098 sequential order. The event payload consists of the logged events and detailed within the event
12099 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)

12100 **Figure D-156– Report Event Configuration Sub-Payload**

12101 **Event ID (mandatory):** The *Event ID* is the attribute ID of the Event Configuration attribute.
12102 Event IDs are detailed in Table D-175 to Table D-183.

12103 **Event Configuration (mandatory):** The configuration bitmap applicable to the event, as defined
12104 in Table D-176.

12105

12106 D.10.2.3.6 GetCIN Command

12107 This command is used to request the ESI to respond with information regarding any pending
12108 change of Customer ID Number.

12109 D.10.2.3.6.1 Payload Details

12110 There are no fields for this command.

12111 D.10.2.3.6.2 Effect on Receipt

12112 The ESI shall send an *UpdateCIN* command. A ZCL Default Response with status
 12113 NOT_FOUND shall be returned if there is no change of Customer ID Number pending.

12114
 12115

12116 D.10.2.4 Commands Generated

12117 Table D-170 lists the cluster specific commands that are generated by the server.

12118 **Table D-170– 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	

12119 D.10.2.4.1 Publish Change of Tenancy Command

12120 This command is used to change the tenancy of a meter.

12121 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)

12122 **Figure D-157– Publish Change of Tenancy Command Payload**12123 D.10.2.4.1.2 Payload Details

12124 **Provider ID (mandatory):** An unsigned 32 bit field containing a unique identifier for the
 12125 commodity provider to whom this command relates.

12126 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. When
 12127 new information is provided that replaces older information for the same time period, this field
 12128 allows devices to determine which information is newer. The value contained in this field is a

12129 unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)
 12130 identifying when the Publish command was issued. Thus, newer information will have a value in
 12131 the *Issuer Event ID* field that is greater than older information.

12132 **Tariff Type (Mandatory):** An 8-bit bitmap identifying the type of tariff published in this
 12133 command. The least significant nibble represents an enumeration of the tariff type as detailed in
 12134 Table D-108 (Generation Meters shall use the ‘Received’ Tariff). The most significant nibble is
 12135 reserved.

12136 **Implementation Date/Time (mandatory):** A UTC Time field to indicate the date from which
 12137 the change of tenancy is to be applied. This value shall always be in advance of the
 12138 *CommandDate/Time* and/or the *LocalTime* by at least 24hrs. An *Implementation Date/Time* of
 12139 0xFFFFFFFF shall cause an existing but pending *Publish Change of Tenancy* command with the
 12140 same *Provider ID* and *Issuer Event ID* to be cancelled.

12141 **Proposed Tenancy Change Control (mandatory):** A 32-bit mask that denotes the functions
 12142 that are required to be carried out on processing of this command. See Table D-163 for further
 12143 details.

12144 D.10.2.4.1.3 When Generated

12145 The *PublishChangeofTenancy* command shall be generated from the ESI, and sent to the meter,
 12146 when a change of tenancy is required. This command can be sent prior to the change of tenancy.
 12147 The meter should use the standard ZCL response.

12148 D.10.2.4.1.4 Effect on Receipt

12149 On receipt of the *PublishChangeofTenancy* command, the device shall update the
 12150 *ChangeofTenancyUpdateDate/Time* and *ProposedTenancyChangeControl* attributes, but only
 12151 action the command at the *ImplementationDate/Time*. At the *ImplementationDate/Time*, the
 12152 device shall check the *ProposedTenancyChangeControl* attribute to understand what additional
 12153 action(s) it must carry out pre and post the change.

12154 D.10.2.4.2 Publish Change of Supplier Command

12155 This command is used to change the Supplier (commodity provider) that is supplying the
 12156 property. This command shall only be used if there is a requirement for the *ProviderID* to be a
 12157 static value within the Prepayment and Price clusters. Should there be a requirement for the
 12158 *ProviderID* to be dynamic, this command and the associated attributes should not be used. It is
 12159 recommended that this command is sent at least one week before the proposed date of change.

12160 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)
------------	-------------------------	---------------------	-----------------	--------------------------	---	-----------------------------	----------------------------	---------------------------------------

12161

Figure D-158 – Publish Change of Supplier Command Payload

12162

D.10.2.4.2.2 Payload Details

12163

Current Provider ID (mandatory): An unsigned 32 bit field containing a unique identifier for the current commodity provider to whom this command relates.

12165

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.

12171

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.

12175

Proposed Provider ID (mandatory): An unsigned 32 bit field containing a unique identifier for the commodity provider associated with the proposed change to the supply. Depending on the *Tariff Type*, this value will be taken from either attribute D.10.2.2.1.4 or D.10.2.2.1.11.

12178

Provider Change Implementation Time (mandatory): A UTC Time field to indicate the date/time at which a proposed change to the provider is to be implemented. Depending on the *Tariff Type*, this value will be taken from either attribute D.10.2.2.1.6 or D.10.2.2.1.13. A *Provider Change Implementation Time* of 0xFFFFFFFF shall cause an existing but pending *Publish Change of Supplier* command with the same *Current Provider ID* and *Issuer Event ID* to be cancelled.

12184

Proposed Provider Name (mandatory): An octet string that denotes the name of the new commodity provider. This is dependent on the *Tariff Type* value; for Received, the parameter should match the attribute in section D.10.2.2.1.12, for all other values it should match the attribute in section D.10.2.2.1.5.

12188

Proposed Provider Contact Details (mandatory): An octet string that denotes the contact details of the new commodity provider. The field shall be capable of storing a 19 character string (the first octet indicates length) encoded in UTF-8 format.

12191 **Provider Change Control (mandatory):** A 32-bit mask that denotes the functions that are
12192 required to be carried out on processing of this command. See section D.10.2.2.1.7 or
12193 D.10.2.2.1.14, depending on the *Tariff Type*.

12194 D.10.2.4.2.3 When Generated

12195 The *PublishChangeofSupplier* command shall be generated from the ESI, and sent to the meter,
12196 when a change of commodity provider is required. It shall also be generated in response to a *Get*
12197 *Change of Supplier* command. The *PublishChangeofSupplier* command contains a start date/time
12198 which allows the command to be sent in advance of the changeover date.

12199 D.10.2.4.2.4 Effect on Receipt

12200 Following receipt of a *PublishChangeofSupplier* command, the meter shall only action the
12201 command at the *ProviderChangeImplementationTime*. At this point in time, the meter shall
12202 check the *Provider Change Control* field to understand what action(s) it must carry out pre and
12203 post the change.

12204

D.10.2.4.3 Request New Password Response Command

12206 This command is used to send the current password to the client. A
12207 *RequestNewPasswordResponse* command is sent either as a response to a *RequestNewPassword*
12208 command or unsolicited when the HES has changed the password.

12209 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)

Figure D-159– RequestNewPasswordResponse Command Payload

12211 D.10.2.4.3.2 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.

12218 **Implementation Date/Time (mandatory):** A UTC Time field to indicate the date at which the
12219 originating command was to be applied.

12220 **Duration in minutes (mandatory):** An unsigned 16-bit integer that denotes the duration in
12221 minutes that the password is valid for. A value of Zero means the password is valid until
12222 changed.

12223 **PasswordType (mandatory):** Indicates which password should be changed. The possible
12224 password types are defined in Table D-171. The password types can be used flexibly by various
12225 end devices. The scope of authority assigned to a password type should be defined by the
12226 corresponding end device.

12227

Table D-171– 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	

12228

12229 **Password (mandatory):** An octet string of length 11 that contains the password (the first octet is
12230 the length, allowing 10 octets for the password).

12231 D.10.2.4.3.3 Effect on Receipt

12232 On receipt of this command, the client shall update the specified password.

12233

12234 D.10.2.4.4 Update SiteID Command

12235 This command is used to set the *SiteID* attribute on a meter (see D.3.2.2.4.8).

12236 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)

12237

Figure D-160—Update SiteID Command Payload

12238

D.10.2.4.4.2 Payload Details

12239 Issuer Event ID (mandatory): Unique identifier generated by the commodity provider. When
12240 new information is provided that replaces older information for the same time period, this field
12241 allows devices to determine which information is newer. The value contained in this field is a

12242 unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)
 12243 identifying when the Publish command was issued. Thus, newer information will have a value in
 12244 the *Issuer Event ID* field that is larger than older information.

12245 **SiteID Time (mandatory):** A UTC Time field to denote the time at which the update of *SiteID*
 12246 will take place. A date/time of 0x00000000 shall indicate that the command should be executed
 12247 immediately (comparison against a time source should NOT be made in this case). A date/time of
 12248 0xFFFFFFFF shall cause an existing but pending *Update SiteID* command with the same
 12249 *Provider ID* and *Issuer Event ID* to be cancelled.

12250 **Provider ID:** An unsigned 32-bit field containing a unique identifier for the commodity provider
 12251 to whom this command relates.

12252 **SiteID (mandatory):** An octet string that denotes the Site ID.

12253

12254 **D.10.2.4.5 SetEventConfiguration Command**

12255 This command provides a method to set the event configuration attributes, held in a client device.

12256 **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)

12257 **Figure D-161– Set Event Configuration Command Payload**

12258 **D.10.2.4.5.2 Payload Details**

12259 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. When
 12260 new information is provided that replaces older information for the same time period, this field
 12261 allows devices to determine which information is newer. The value contained in this field is a
 12262 unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)
 12263 identifying when the command was issued. Thus, newer information will have a value in the
 12264 *Issuer Event ID* field that is larger than older information.

12265 **Start Date/Time (mandatory):** A UTC Time field to indicate the date and time at which the
 12266 new configuration is to be applied.

12267 **Event Configuration (mandatory):** This field holds the new event configuration to be applied,
 12268 as defined in Table D-176.

Configuration Control (mandatory): The *Configuration Control* enumeration allows the new configuration value to be applied to several events via a single command. The value of this field defines the format of the event configuration payload:

Table D-172– 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 – OFF	Reserved

D.10.2.4.5.2.1 Apply by List

12275 The ‘Apply by List’ option allows individual or lists of events to be configured by a single
12276 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)

Figure D-162– ‘Apply by List’ Sub-Payload

12278 **Number of Events (mandatory):** This field holds the number of events contained within the
12279 command.

12280 **Event ID (mandatory):** The *Event ID* is the attribute ID of the event configuration attribute.
12281 Event IDs are detailed in Table D-175 to Table D-183.

12282 D.10.2.4.5.2.2 Apply by Event Group

The ‘Apply by Event Group’ option allows all events belonging to a stated event group (attribute set) to be configured by a single command:

Octets	2
Data Type	Unsigned 16 bit Integer
Field Name	Event Group ID (M)

Figure D-163– ‘Apply by Event Group’ Sub-Payload

Event Group ID (mandatory): The *Event Group ID* field indicates which attribute set the event belongs to (see Table D-173). The *Event Group ID* is in the form ‘0xnnFF’, where *nn* is the Attribute Set Identifier (the final attribute in the sets defined in Table D-175 to Table D-183 is reserved as a ‘wildcard’ attribute to allow definition of the *Event Group IDs*).

12290 **D.10.2.4.5.2.3 Apply by Log Type**

12291 The ‘Apply by Log Type’ option allows all configurations recorded in a given log to be
 12292 configured:

Octets	1
Data Type	Unsigned 8 bit Integer
Field Name	Log ID

12293 **Figure D-164– ‘Apply by Log Type’ Sub-Payload**

12294 **Log ID:** The *Log ID* specifies the log ID of events to be updated with the new *Configuration*
 12295 *Value* field passed in the command. The applicable values for this field are defined by bits 0-2 of
 12296 the Table D-176.

12297 **D.10.2.4.5.2.4 Apply by Configuration Match**

12298 The ‘Apply by Configuration Match’ option allows all events matching a given configuration
 12299 value to be changed to the new configuration value:

Octets	1
Data Type	8-bit Bitmap
Field Name	Configuration Value Match (M)

12300 **Figure D-165– ‘Apply by Configuration Match’ Sub-Payload**

12301 **Configuration Value Match (mandatory):** This field indicates that any configuration attribute
 12302 which matches this value shall be assigned the new configuration value passed in the *Event*
 12303 *Configuration* field of the main command payload (see D.10.2.4.5.1).

12304 **D.10.2.4.6 GetEventConfiguration Command**

12305 This command allows the server to request details of event configurations.

12306 **D.10.2.4.6.1 Payload Format**

Octets	2
Data Type	Unsigned 16-bit Integer
Field Name	Event ID (M)

12307 **Figure D-166– Get Event Configuration Command**12308 **D.10.2.4.6.2 Payload Details**

12309 **Event ID (mandatory):** The *Event ID* specifies a particular event to be queried. A value of
 12310 0xFFFF is reserved to indicate all event IDs. A value equal to the *Event Group ID* (the final
 12311 attribute in the sets defined in Table D-175 to Table D-183 is reserved for this purpose) shall
 12312 indicate all event IDs within the indicated attribute set. The Event IDs are detailed in Table
 12313 D-175 to Table D-183.

12314

12315 D.10.2.4.7 **Update CIN Command**12316 This command is used to set the *CustomerIDNumber* attribute held in the Metering cluster (see
12317 D.3.2.2.4.18).12318 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)

12319

Figure D-167– Update CIN Command Payload12320 D.10.2.4.7.2 **Payload Details**12321 **Issuer Event ID (mandatory):** Unique identifier generated by the commodity provider. When
12322 new information is provided that replaces older information for the same time period, this field
12323 allows devices to determine which information is newer. The value contained in this field is a
12324 unique number managed by upstream servers or a UTC based time stamp (UTCTime data type)
12325 identifying when the Publish command was issued. Thus, newer information will have a value in
12326 the *Issuer Event ID* field that is larger than older information.12327 **CIN Implementation Time (mandatory):** A UTC Time field to denote the date/time at which
12328 the updated *CustomerIDNumber* will become active. A value of 0x00000000 shall indicate that
12329 the command should be executed immediately (comparison against a time source should NOT be
12330 made in this case). A value of 0xFFFFFFFF shall cause an existing but pending *UpdateCIN*
12331 command with the same *Provider ID* and *Issuer Event ID* to be cancelled.12332 **Provider ID (mandatory):** An unsigned 32-bit field containing a unique identifier for the
12333 commodity provider to whom this command relates.12334 **CustomerIDNumber (mandatory):** An octet string that denotes the Customer ID Number.

12335

12336 D.10.2.4.7.3 **Effect on Receipt**12337 Upon successful receipt of this command, the meter shall update the *CustomerIDNumber*
12338 attribute and return a ZCL Default Response indicating SUCCESS.12339 A ZCL Default Response indicating NOT_AUTHORIZED shall be returned if the Provider ID
12340 contained within the command does not match the current Provider ID. For all other failures, a
12341 ZCL Default Response indicating FAILURE shall be returned.

12342

12343

12344 **D.10.3 Client**12345 **D.10.3.1 Dependencies**

- 12346 • Support for ZCL Data Types

- 12347 • Events carried using this cluster include a timestamp with the assumption that target devices
-
- 12348 maintain a real time clock. Devices can acquire and synchronize their internal clocks via the
-
- 12349 ZCL Time server.

12350 **D.10.3.2 Attributes**12351 **Table D-173– 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 – 0xFF	Reserved

12352

12353 **D.10.3.2.1 Supplier Attribute Set**

12354

Table D-174– 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					

12355

12356 D.10.3.2.1.1 ProviderID Attribute

12357 An unsigned 32 bit field containing a unique identifier for the commodity provider to whom this
 12358 attribute relates.

12359 D.10.3.2.1.2 ReceivedProviderID Attribute

12360 An unsigned 32 bit field containing a unique identifier for the commodity provider to whom this
 12361 attribute relates. This attribute is only for the Received supply.

12362

12363 D.10.3.2.2 **Price Event Configuration Attribute Set**

12364 The following attributes allow events related to pricing to be configured.

12365 It should be noted that triggers for events are an implementation issue, however it is suggested
 12366 that the ‘Tariff Activated’ events should only be logged (if configured to do so) when moving
 12367 from one tariff type to another, not when a tariff is modified.

12368

Table D-175– 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

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory/Optional
0x010B	PriceMatrixPassiveUpdated	8-bit Bitmap	0x00 – 0xFF	Read only		O
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

12369

12370 D.10.3.2.2.1 Event Configuration Attributes12371 The least-significant 3 bits of the Event Configuration bitmaps indicate how the event should be
12372 logged; the remaining bits provide options for treatment rules to be applied.

12373

Table D-176– 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	

12374

12375 D.10.3.2.3 **Metering Event Configuration Attribute Set**

12376 The following attributes allow events related to the meter to be configured.

12377

Table D-177– 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
	<i>Electricity: Power Failure</i>					
	<i>Gas: Not Defined</i>					
	<i>Water: Pipe Empty</i>					
	<i>Heat/Cooling: Temperature Sensor</i>					
0x0204	Supply Quality	8-bit Bitmap	0x00 – 0xFF	Read only		O
	<i>Electricity: Power Quality</i>					
	<i>Gas: Low Pressure</i>					
	<i>Water: Low Pressure</i>					
	<i>Heat/Cooling: Burst Detect</i>					

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
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
0x0217	OverCurrentL2	8-bit Bitmap	0x00 – 0xFF	Read only		O

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
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
0x0231	PressureTooLow	8-bit Bitmap	0x00 – 0xFF	Read only		O
0x0232	PressureTooHigh	8-bit Bitmap	0x00 – 0xFF	Read only		O

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
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
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

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
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
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

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
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
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					

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
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
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

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
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
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

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
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
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

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
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

12378

12379 D.10.3.2.3.1 Event Configuration Attributes

12380 The attributes in this set allow a server device to configure how an event is handled when
 12381 triggered. All attributes in this set are bitmaps as defined in Table D-176.

12382 D.10.3.2.4 Messaging Event Configuration Attribute Set

12383 The following attributes allow events related to messaging to be configured.

12384

Table D-178– 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

12385

12386 D.10.3.2.4.1 Event Configuration Attributes12387 The attributes in this set allow a server device to configure how an event is handled when
12388 triggered. All attributes in this set are bitmaps as defined in Table D-176.12389 D.10.3.2.5 **Prepayment Event Configuration Attribute Set**

12390 The following attributes allow events related to prepayment to be configured.

12391

Table D-179– 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
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

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
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
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

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
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
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

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
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
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

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
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

12392

12393 D.10.3.2.5.1 Event Configuration Attributes

12394 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-176.

12396 D.10.3.2.6 Calendar Event Configuration Attribute Set

12397 The following attributes allow events related to calendars to be configured.

12398

Table D-180– 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

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
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
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

12399

12400 D.10.3.2.6.1 Event Configuration Attributes12401 The attributes in this set allow a server device to configure how an event is handled when
12402 triggered. All attributes in this set are bitmaps as defined in Table D-176.12403 D.10.3.2.7 Device Management Event Configuration Attribute Set

12404 The following attributes allow events related to device management to be configured.

12405

Table D-181– Device Management Event Configuration Attribute Set

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
----------------------	------	------	-------	--------	---------	----------------------

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
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

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
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
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

12406

12407 D.10.3.2.7.1 Event Configuration Attributes

12408 The attributes in this set allow a server device to configure how an event is handled when
12409 triggered. All attributes in this set are bitmaps as defined in Table D-176.

12410 D.10.3.2.8 Tunnel Event Configuration Attribute Set

12411 The following attributes allow events related to tunneling to be configured.

12412

Table D-182– 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

12413

12414 D.10.3.2.8.1 Event Configuration Attributes12415 The attributes in this set allow a server device to configure how an event is handled when
12416 triggered. All attributes in this set are bitmaps as defined in Table D-176.12417 D.10.3.2.9 OTA Event Configuration Attribute Set

12418 The following attributes allow events related to OTA to be configured.

Table D-183– 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					
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

Attribute Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
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

12420

12421 D.10.3.2.9.1 Event Configuration Attributes12422 The attributes in this set allow a server device to configure how an event is handled when
12423 triggered. All attributes in this set are bitmaps as defined in Table D-176.

12424

12425 D.10.3.3 Commands Received

12426 The client receives the cluster-specific response commands detailed in sub-clause D.10.2.4.

12427 D.10.3.4 Commands Generated

12428 The client generates the cluster-specific commands detailed in sub-clause D.10.2.3, as required by
12429 the application.

12430

12431 D.10.4 Application Guidelines

12432 D.10.4.1 Passwords

12433 The use of Password within this cluster could also be viewed as PIN codes. The current use case
12434 for Passwords is to cover either the consumer PIN code, or to secure the engineer maintenance
12435 screens found on a metering device.

12436 D.10.4.2 Consumer Password Use Case

12437 The Password or (normally) PIN code is part of the application and, as such, not a data item that
12438 would need to be supplied by the HES or held by the server. There is normally a screen on a
12439 device to be able to set or enter a new password. The main use case for the consumer is therefore
12440 to instruct the device to reset the PIN so that the consumer can again gain access to the IHD
12441 screens. The server is therefore only required to hold an access list that contains the device
12442 EUI64 address.

12443 D.10.4.3 Engineer Password Use Case

12444 The Engineer password is normally used to access maintenance screens on meters, so that key
12445 functions can be secured and only accessed by an authorised personal, or for data that is sensitive
12446 to the operation of the device such as Joining or Leaving the HAN.

12447 D.10.4.4 Password Security Recommendations

12448 If additional security is required by the application, it is recommended that the password octet-
12449 string is sent as a hashed value, using MMO hashing to create a 48-bit hashed value. The Device
12450 Management Cluster is APS Secured and the password is always unicast to the individual device.
12451 An Access Control List within the server is recommended, to allow for the management of the
12452 passwords against the EUI-64 address of the device. However, the actual format of the password
12453 octet-string is down to the implementation requirements of the system.

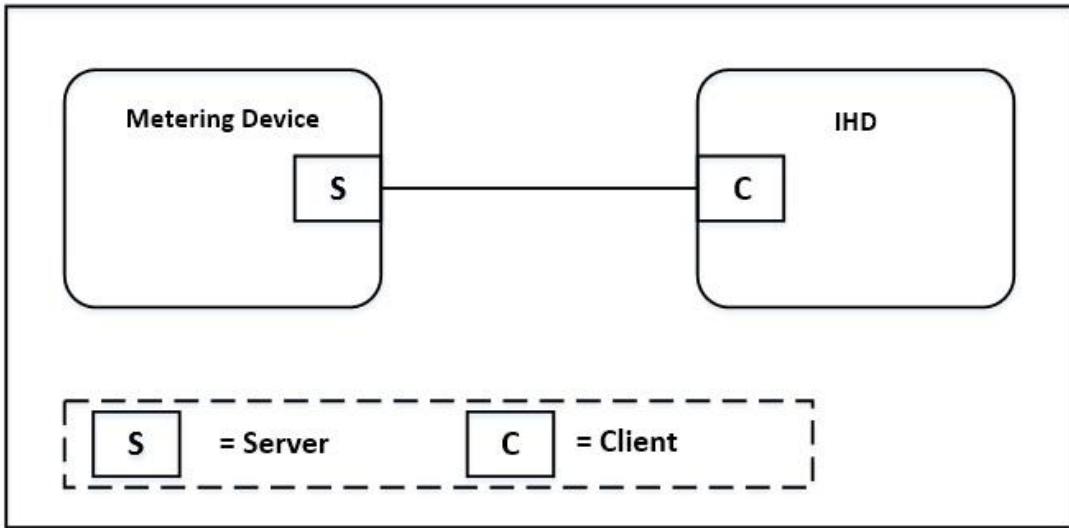
12454 If a device is unable to use a password then NO data should be shown that has been deemed to be
12455 password protected.

12456

12457 **D.11 Events Cluster**

12458 **D.11.1 Overview**

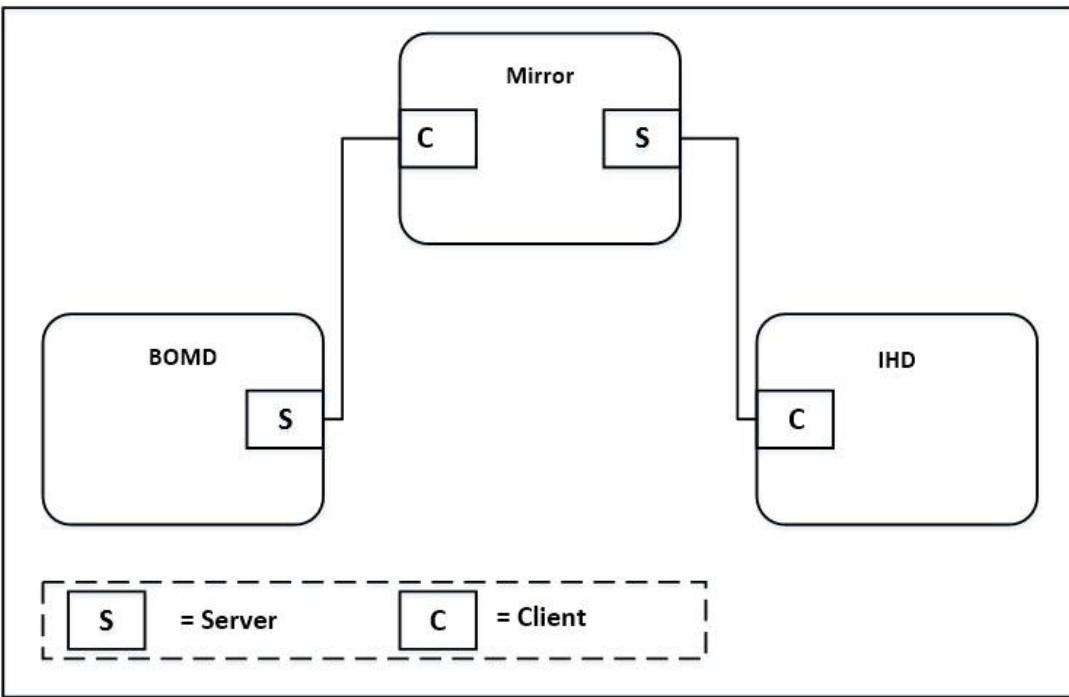
12459 This cluster provides an interface for passing event information between ZigBee devices. Events
12460 are generated and logged by a server device and read by a client device.



12461

12462

Figure D-168– Event Cluster Client Server Example



12463

12464

Figure D-169– Mirrored BOMD Event Cluster Client Server Example

12465

12466 D.11.2 Server

12467 D.11.2.1 Dependencies

- Events carried using this cluster include a timestamp with the assumption that target devices maintain a real time clock. Devices can acquire and synchronize their internal clocks via the ZCL Time server.
 - A server device supporting this cluster should also support the Device Management Cluster in order to allow events to be configured over the air.
 - In order that Events Cluster client devices are able to receive events published from an Events Cluster server on a BOMD, the BOMD mirror should support both an Events cluster client and server. The BOMD should publish events to the mirror and the mirror should, if required (based on the control flags in the *PublishEvent* command), publish events to all bound Events Cluster client devices.
 - Events Cluster client devices wishing to receive events published from a BOMD shall bind to the Events Cluster server on the BOMD mirror.
 - A Mirror may store the events pushed from a BOMD, effectively mirroring the BOMD event logs. The Mirror may also support the reading and clearing of event logs by Events Cluster client devices.
 - How events are internally stored within an Events Clusters server device is out of scope of this specification.

12485

12486 D.11.2.2Attributes

12487 None

12488 D.11.2.3 Commands Received

12489 Table D-184 lists the cluster specific commands that are received by the server.

12490

Table D-184– Cluster Specific Commands Received by the Server

Command Identifier Field Value	Description	Mandatory / Optional
0x00	GetEventLog	O

0x01	Clear Event Log Request	O
------	-------------------------	---

12491

12492 D.11.2.3.1 Get Event Log Command12493 The *GetEventLog* command allows a client to request events from a server's event logs. One or
12494 more *PublishEventLog* commands are returned on receipt of this command.12495 The *LogID* sub-field, in conjunction with the *Event ID* field, shall provide the filtering to enable
12496 the desired event(s) to be identified. The following examples show the usage of these 2 fields:-

12497 1 Get all events from the Security Event Log (Log ID = Security (4), Event ID = 0x0000)

12498 2 Get all events from all logs (Log ID = 0, Event ID = 0x0000)

12499 3 Get all occurrences of a specific event 0x1111 from all logs (Log ID = 0, Event ID =
12500 0x1111)12501 4 Get all occurrences of a specific event 0x1111 from the Security Event log (Log ID =
12502 Security (4), Event ID = 0x1111).

12503

12504 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)

12505 Figure D-170– Get Event Log Command Payload

12506 D.11.2.3.1.2 Payload Details12507 **Event Control/Log ID (mandatory):** The least significant nibble is an enumeration indicating
12508 the Log ID (see Table D-185). The most significant nibble is a bitmap indicating control options
12509 (see Table D-186):

12510 Table D-185– Log ID Enumeration

Bit	Description
-----	-------------

	Enumerated Value	Description
0-3	0x0	All logs
	0x1	Tamper Log
	0x2	Fault Log
	0x3	General Event Log
	0x4	Security Event Log
	0x5	Network Event Log
	0x6-0xF	Reserved

12511

12512

Table D-186– 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

12513

12514 **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-175 to Table D-183.

12517 *Note: If event configuration is supported via the device management cluster the Event IDs are defined in tables Table D-175 to Table D-183.*

12519 **Start Time (mandatory):** This field specifies the start time (earliest time) of the range of events to be returned. Events that match the search criteria and have a timestamp **greater than or equal to** the start time shall be returned.

12522 **End Time (mandatory):** specifies the end time (latest time) of the range of events to be reported in the response. Events that match the search criteria and have a timestamp **less than** the specified end time shall be returned. Events with a timestamp **equal to** that of the *End Time* shall not be returned; this ensures that, in the case where the *End Time* is set to the current time, events generated whilst reading the event log are not included in the response.

12527 **Number of Events (mandatory):** This parameter indicates the maximum number of events requested i.e. the maximum number of events that the client is willing to receive; the value 0x00 indicates all events that fall into the defined criteria.

12530 **Event Offset (mandatory):** The *Event Offset* field provides a mechanism to allow client devices to page through multiple events which match a given search criteria. As an example, a client device requests two events from a given search criteria with an *Event Offset* of 0. The server returns the two most recent events (events 1 and 2) in a *PublishEvent* command and indicates

12534 that 4 events match the given criteria. The client re-sending the original *Get Event Log* command,
 12535 but with the *Event Offset* field now set to 2, shall result in the server returning events 3 and 4.

12536 D.11.2.3.1.3 Effect on Receipt

12537 On receipt of this command, the device shall respond with a *PublishEventLog* command A ZCL
 12538 Default Response with status NOT_FOUND shall be returned if no events match the given
 12539 search criteria.

12540

12541 D.11.2.3.2 Clear Event Log Request Command

12542 This command requests that an Events server device clear the specified event log(s). The Events
 12543 server device SHOULD clear the requested events logs, however it is understood that market
 12544 specific restrictions may be applied to prevent this.

12545 D.11.2.3.2.1 Payload Format

Octets	1
Data Type	8-bit BitMap
Field Name	Log ID (M)

12546 **Figure D-171– Clear Event Log Request Command Payload**

12547

12548 D.11.2.3.2.2 Payload Details

12549 **Log ID (mandatory):** The least significant nibble specifies the Log to be cleared (see Table
 12550 D-185). The most significant nibble is reserved.

12551 D.11.2.3.2.3 Effect on Receipt

12552 On receipt of this command, a device supporting the Events cluster as a server should clear the
 12553 specified event logs. A *Clear Event Log Response* command shall be generated, indicating which
 12554 event logs have been successfully cleared.

12555

12556 D.11.2.4 Commands Generated

12557 Table D-187 lists the cluster specific commands that are generated by the server.

12558 **Table D-187– 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

12559

12560 D.11.2.4.1 Publish Event Command

12561 This command is generated upon an event trigger from within the reporting device and if enabled
 12562 by the associated Event Configuration (bitmap) attribute in the Device Management cluster (see
 12563 Table D-176 for further information).

12564 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)

12565 **Figure D-172– Publish Event Command Payload**12566 D.11.2.4.1.2 Payload Details

12567 **Log ID (mandatory):** The least significant nibble is an enumeration indicating the Log ID (see
 12568 Table D-185). The most significant nibble is reserved.

12569 **Event ID (mandatory):** The *Event ID* specifies a particular event. If event configuration is
 12570 supported (via the Device Management cluster), the Event IDs are as defined in Table D-175 to
 12571 Table D-183.

12572 **Event Time (mandatory):** The timestamp of the event occurrence in UTC format.

12573 **Event Control (mandatory):** An 8-bit bitmap specifying actions to be taken regarding this
 12574 event:

12575 **Table D-188– 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

12576

12577 **Event Data (mandatory):** A variable length octet string array used to hold additional
 12578 information captured when the event occurred. The first element (element 0) of the array
 12579 indicates the length of the string, NOT including the first element.

12580

12581 **D.11.2.4.2 Publish Event Log Command**

12582 This command is generated on receipt of a *Get Event Log* command. The command shall return
 12583 the most recent event first, up to the number of events requested.

12584 **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)

12585 **Figure D-173– Publish Event Log Command Payload**

12586 **D.11.2.4.2.2 Payload Details**

12587 **Total Number of Matching Events (mandatory):** This field indicates the total number of
 12588 events found which match the search criteria received in the associated *Get Event Log* command.
 12589 The value of this field may be greater than the total number of events requested; if this is the case
 12590 then further events may be retrieved using the *Event Offset* field of the *Get Event Log* command
 12591 (see D.11.2.3.1).

12592 **Command Index (mandatory):** In the case where the entire number of events being returned
 12593 does not fit into a single message, the *Command Index* is used to count the required number of
 12594 *Publish Event Log* commands. The *Command Index* starts at 0 and is incremented for each
 12595 command returned due to the same *Get Event Log* command.

12596 **Total Commands (mandatory):** This parameter indicates the total number of *Publish Event Log*
 12597 commands that are required to return the requested event logs.

12598 **D.11.2.4.2.2.1 Log Payload**

12599 The *Log Payload* is a series of events and associated data. The event payload consists of the
 12600 logged events as detailed in Figure D-174:

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

Octets	1	1	2	4	1..255	...	1	2	4	1..255
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)

12601

Figure D-174– Publish Event Log Sub-Payload

12602 **Number of Events /Log Payload Control (mandatory):** This field is split into two parts; the least significant nibble represents the *Log Payload Control* as defined in Table D-189, whilst the most significant nibble indicates the number of events contained within the log payload of this command. Note that an event which crosses a payload boundary is considered to be 1 event in the log payload. Wherever possible events SHOULD NOT be sent across payload boundaries.

12607

Table D-189– 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

12608

12609 **Log ID (mandatory):** The least significant nibble is an enumeration indicating the Log ID (see Table D-185). The most significant nibble is reserved.

12611 **Event ID:** The *Event ID* specifies a particular event. If event configuration is supported (via the Device Management cluster), the Event IDs are as defined in Table D-175 to Table D-183).

12613 **Event Time:** The timestamp of the event occurrence in UTC format.

12614 **Event Data:** A variable length octet string array used to hold additional information captured when the event occurred. The first element (element 0) of the array indicates the length of the string, NOT including the first element. This field should contain a single octet of 0x00 when ‘minimal information’ is requested in the associated *Get Event Log* command (see D.11.2.3.1.2 for further details).

12619

12620 D.11.2.4.3 Clear Event Log Response Command

12621 This command is generated on receipt of a *Clear Event Log Request* command.

12622 D.11.2.4.3.1 Payload Format

Octets	1
--------	---

Data Type	8 Bit Bitmask
Field Name	ClearedEventsLogs (M)

12623

Figure D-175– Clear Event Log Response Command Payload12624 D.11.2.4.3.2 Payload Details12625 **ClearedEventsLogs (mandatory):** This 8-bit BitMask indicates which logs have been cleared,
12626 as detailed in Table D-187.12627 *Note: It is understood that certain markets may require that event logs cannot be cleared; this*
12628 *BitMask provides a method for the server device to indicate which logs have been successfully*
12629 *cleared.*

12630

Table D-190– 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

12631

12632

12633 **D.11.3 Client**12634 **D.11.3.1 Dependencies**12635 Events carried using this cluster include a timestamp with the assumption that target devices
12636 maintain a real time clock. Devices can acquire and synchronize their internal clocks via the ZCL
12637 Time server.12638 **D.11.3.2 Attributes**

12639 None.

12640 [D.11.3.3 Commands Received](#)

12641 See section D.11.2.4.

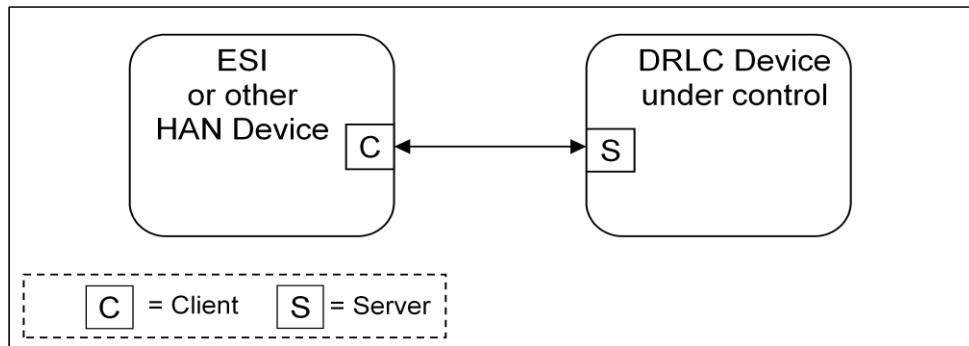
12642 [D.11.3.4 Commands Generated](#)

12643 See section D.11.2.3.

12644

12645 **D.12 Energy Management Cluster**12646 **D.12.1 Overview**

12647 This cluster provides a way of modifying DRLC events, energy consumption behaviour and
 12648 querying the status of DRLC events.

12649
12650 **Figure D-176– Energy Management Cluster**
12651

12652 Note that the ESI is defined as the Client. The DRLC device is a Server in this case; it holds the
 12653 attributes and receives commands.
 12654

12655 **D.12.2 Server**12656 **D.12.2.1 Dependencies**

12657 A server device shall support the DRLC cluster as a client.

12658 **D.12.2.2 Attributes**12659 **Table D-191– 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	<i>CurrentEventID</i>	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

12660

12661 D.12.2.2.1 **Load Control State Attribute**12662 This attribute shall be a BitMap showing the current state of the device. The attribute shall be
12663 read-only.

12664

Table D-192– 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

12665

12666 **Relay Open or Consumption Interrupted:** The device being controlled has been turned off and
12667 prevented from consuming electric power or another commodity, either by opening a relay or
12668 some other means.12669 **Event In Progress:** There is an event in progress. The current time is between the Effective Start
12670 Time and Effective End Time of the event. When this bit is set, the *CurrentEventStatus* attribute
12671 is valid.12672 **Power Stabilizing:** The device has automatically reduced consumption of the commodity for an
12673 automatic reason, to aid in the stability of the system. Possible actions and reasons might be:

- 12674 • A random wait after a power outage before starting an electric motor.

- 12675 • Duty cycling heating after a long power outage, to prevent all electric heaters being on at the
12676 same time.
- 12677 • Turning off a furnace if the gas pressure drops, open electric relay if the frequency or voltage
12678 are off, etc.

12679 **Other Load Reduction:** This bit indicates the device has automatically reduced consumption of
12680 the commodity for some non-consumer-initiated reason. Possible automatic actions and reasons
12681 might be:

- 12682 • A PCT raising the cooling set point in response to an increase in price.
- 12683 • A Load Control Device turning off a pool pump or other non-essential appliance when
12684 energy credits are low.

12685 **Current Flow or Consuming Commodity:** This bit indicates that the device is currently
12686 consuming the commodity. The bit not being set means either no commodity being consumption
12687 or the device does not have the ability to detect consumption. Support is optional. The bit shall be
12688 cleared if not supported.

12689 **Load Call:** This bit is set if there is currently no consumption but the device under control would
12690 consume power if able to. Support is optional. The bit shall be cleared if not supported.

12691

12692 D.12.2.2 CurrentEventID Attribute

12693 If an event is in progress (current time is between the Effective Start Time and Effective End
12694 Time of an event), this attribute SHALL indicate the *Issuer Event ID* of the active event. The
12695 invalid Event ID 0xFFFFFFFF SHALL be used when an event is NOT active. The attribute
12696 SHALL be read-only.

12697 D.12.2.2.3 CurrentEventStatus Attribute

12698 This attribute represents the value returned in the *Event Control* field of the latest *Report Event*
12699 *Status* command (see D.2.3.3.1). This attribute provides a mechanism to allow a remote device to
12700 query whether this client device is currently participating in a load control event. Typical use of
12701 this attribute is expected to be for the refresh of UIs. The attribute SHALL be read-only.

12702 In conjunction with this new attribute, the BitMap defined for the *Event Control* field of both
12703 *Load Control Event* and *Report Event Status* commands SHALL be extended to cater for the
12704 following additional control options (note that only *Report Event Status* commands will use these
12705 additional bits):

12706

12707

Table D-193– 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

12708

12709 **Randomized Start Time Attribute:** Set if the current event had a randomized start time.12710 **Randomized Duration:** Set if the duration of the current event is randomized.12711 **Extended Bits Present:** This bit will always return 1. This allows the field to be used in DRLC
12712 *Event Status* messages.12713 **Event Active:** Set if the current time lies between the Effective Start Time and Effective End
12714 Time of the event identified by the *CurrentEventID* attribute (Note that, if this bit is not set when
12715 the *CurrentEventStatus* attribute is read, none of the other bits are valid since there is no current
12716 event).12717 **Device Participating in Event:** Set if the device is (or will be when the event starts)
12718 participating in the event (i.e. not opted out) Note that a device can participate in an event and
12719 not actually do anything, i.e. this bit could be set while the *Load Reduction* bit is not.12720 **Reducing Load:** Set if the device is currently shedding load in response to a DR event. Set if an
12721 active load control event is duty cycling and currently off, or if the new set points or offsets are
12722 lowering demand. Clear if the device is participating in an event, but is currently in the On
12723 portion of the duty cycle. Clear if none of the fields of the DRLC event applied to the device (e.g.
12724 only setpoints were set but the device is a pool pump). Support is optional. The bit shall be set if
12725 not supported.12726 **On at End of Event:** Set if the device will return to using a normal load after the event has
12727 completed. For example, this would be False if the device supported the On/Off cluster ([B1])
12728 and the *OnOff* attribute was set to OFF. It would also be False if the device was a PCT, the
12729 current temperature was lower than the set point and the device was in cooling mode. Note that,
12730 if an event is not active, this value may not be reliable. Support is optional; if unsupported, the bit
12731 shall be set.

12732 D.12.2.2.4 **Conformance Level Attribute**

12733 This is the minimum criticality level of a DRLC event that the device will observe. Events with a
12734 criticality level lower than the *Conformance Level* will be opted out. If a change in the
12735 *Conformance Level* results in a currently running event being opted out (or in), a
12736 *ReportEventStatus* message must be generated. If the event(s) has(have) not started, the device
12737 may send a *ReportEventStatus* immediately or when the event starts. The attribute shall be
12738 read/write if supported, with a valid range of 1 -7, or 0 and read-only if unsupported.

12739 D.12.2.2.5 **MinimumOffTime, MinimumONTime & MinimumCyclePeriod Attributes**

12740 These attributes are measured in seconds and used to determine the duty cycling times. The
12741 *MinimumOffTime* and *MinimumOnTime* may also be enforced when starting or ending a DRLC
12742 event, or when the On/Off cluster opens or closes the relay. From these values, the DutyOnTime
12743 and DutyOffTime times are calculated.

12744 *MinimumCyclePeriod* is the shortest cycling period allowed for duty cycling, while
12745 *MinimumOnTime* and *MinimumOffTime* are the shortest times the device should be allowed on or
12746 off. The on/off minimums are to protect devices that can be damaged by being on or off for short
12747 time periods, while the *MinimumCyclePeriod* prevents the relay from cycling too quickly. The
12748 attribute values shall be 0xFFFF and read-only if unsupported, or read/write if supported. A
12749 *MinimumCyclePeriod* of 0 indicates that duty cycling shall be disabled; an event calling for any
12750 duty cycling will then just turn the device off for the duration of the event.

12751

12752 **Algorithm for calculating DutyOnTime and DutyOffTime**

12753 DutyOnTime: the amount of time in a duty cycle period that the device is on
12754 DutyOffTime:the amount of time in a duty cycle period that the device is off

12755 DutyOnTime = MinimumCyclePeriod * dutyCycleApplied / 100
12756 DutyOffTime = MinimumCyclePeriod - DutyOnTime
12757 //NOTE except for dutyCycleApplied near 99% or 1% we are likely done
12758
12759 If (DutyOnTime < MinimumONTime)
12760 DutyOnTime = MinimumONTime
12761 DutyOffTime = MinimumONTime * (100 - dutyCycleApplied) / dutyCycleApplied
12762
12763 If (DutyOffTime < MinimumOffTime)
12764 DutyOffTime = MinimumOffTime
12765 DutyONTIme = MinimumOffTime * dutyCycleApplied/ (100 - dutyCycleApplied)

12766

12767

12768

12771 D.12.2.3 Attribute Reporting

12772 Attribute reporting is not expected to be used for this cluster.

12773 D.12.2.4 Commands Received

12774 The command IDs received by the Energy Management cluster server are listed in Table D-194.

12775 Table D-194– Received Command IDs for the Energy Management Server

Command Identifier Field Value	Description	Mandatory/Optional
0x00	<i>ManageEvent</i>	M

12776

12777 D.12.2.4.1 Manage Event Command

12778 The *Manage Event* command allows a remote device (such as an IHD or web portal) to change
12779 the behavior of a DRLC cluster client when responding to a DRLC *Load Control Event*.

12780 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

12781 Figure D-177– *Manage Event Command Payload*12782 D.12.2.4.1.2 Payload Details

12783 **Issuer Event ID (mandatory):** Unique identifier generated by the Energy provider. The value of
12784 this field allows the command to be matched with a specific Load Control Event. A value of
12785 0xFFFFFFFF indicates the current running event.

12786 **Device Class (mandatory):** Bit encoded field representing the DRLC client Device Class to
12787 apply the current *Load Control Event*. Each bit, if set individually or in combination, indicates
12788 the class of device(s) the *Manage Event* command is directed at. If the requested device class(es)
12789 is not implemented by the receiving device, the command should be ignored.

12790 **Utility Enrollment Group (UEG) (mandatory):** The DRLC client *Utility Enrollment Group*
12791 field can be used in conjunction with the *Device Class* bits. It provides a mechanism to direct the
12792 *Manage Event* command to groups of devices. If the requested UEG is non-zero and does not
12793 match the UEG of the receiving device, the command should be ignored.

12794 **Action(s) Required:** Bit encoded field indicating the action(s) to be carried out in regard to the
 12795 associated event. Bits, set individually or in combination, shall be provided for the following
 12796 actions:

12797

Table D-195– 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

12798

12799 If the *Manage Event* command is valid, a *Report Event Status* command will be returned
 12800 regardless of whether any bits are set.

12801 Notes:

- 12802 **1** If the load control event is mandatory and the device is Smart Energy compliant, requests to
 12803 opt-out shall be ignored. Opt-in shall always be honored.
- 12804 **2** If the device does not support duty cycling, or the load control event did not request duty
 12805 cycling, requests to modify duty cycling should be ignored, however a *Report Event Status*
 12806 command should still be sent.
- 12807 **3** If the *ManageEvent* command action was to opt-out or opt-in, the *Event Status* field in the
 12808 associated *Report Event Status* response shall be 0x04 (User Opt-out) or 0x05 (User Opt-in).
 12809 For all other actions, the *Event Status* field in the associated *Report Event Status* response
 12810 shall be set to 0x01 (if the load control event has not yet started), 0x02 (if the event is
 12811 running and opted in) or 0x04 (if the event is running but currently opted out; this is to allow
 12812 for backwards compatibility).

12813 The rationale behind disabling duty cycling is to give the consumer more control over how much
 12814 energy to shed during the event, without necessarily requiring them to place their device in a
 12815 state where it will be off after the event. The response to disabling duty cycling will be seen as a
 12816 value of 0 in the duty cycle applied field of the *Report Event Status* message.

12817

12818 D.12.2.5 Commands Generated

12819 The command IDs generated by the Energy Management cluster server are listed in Table
 12820 D-196.

12821	Table D-196– Generated Command IDs for the Energy Management Server																						
	<table border="1"> <thead> <tr> <th>Command Identifier Field Value</th><th>Description</th><th>Mandatory/Optional</th></tr> </thead> <tbody> <tr> <td>0x00</td><td><i>ReportEventStatus</i></td><td>M</td></tr> </tbody> </table>	Command Identifier Field Value	Description	Mandatory/Optional	0x00	<i>ReportEventStatus</i>	M																
Command Identifier Field Value	Description	Mandatory/Optional																					
0x00	<i>ReportEventStatus</i>	M																					
12822																							
12823	D.12.2.5.1 Report Event Status Command																						
12824	This command is reused from the DRLC cluster. This command is generated in response to the <i>Manage Event</i> command. It is likely that a <i>Manage Event</i> command will also change a DRLC event causing the DRLC client to send a DRLC <i>Report Event Status</i> message as well (see D.2.3.3.1).																						
12825																							
12826																							
12827																							
12828	D.12.2.5.1.1 Payload Format																						
12829	The <i>Report Event Status</i> command payload shall be formatted as illustrated in Figure D-178.																						
12830	<table border="1"> <thead> <tr> <th>Octets</th><th>4</th><th>1</th><th>4</th><th>1</th><th>2</th><th>2</th></tr> </thead> <tbody> <tr> <td>Data Type</td><td>Unsigned 32-bit Integer</td><td>Unsigned 8-bit Integer</td><td>UTCTime</td><td>Unsigned 8-bit Integer</td><td>Unsigned 16-bit Integer</td><td>Unsigned 16-bit Integer</td></tr> <tr> <td>Field Name</td><td>Issuer Event ID (M)</td><td>Event Status (M)</td><td>Event Status Time (M)</td><td>Criticality Level Applied (M)</td><td>Cooling Temperature Set Point Applied (O)</td><td>Heating Temperature Set Point Applied (O)</td></tr> </tbody> </table>		Octets	4	1	4	1	2	2	Data Type	Unsigned 32-bit Integer	Unsigned 8-bit Integer	UTCTime	Unsigned 8-bit Integer	Unsigned 16-bit Integer	Unsigned 16-bit Integer	Field Name	Issuer Event ID (M)	Event Status (M)	Event Status Time (M)	Criticality Level Applied (M)	Cooling Temperature Set Point Applied (O)	Heating Temperature Set Point Applied (O)
Octets	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)																	
12831	<table border="1"> <thead> <tr> <th>Octets</th><th>1</th><th>1</th><th>1</th></tr> </thead> <tbody> <tr> <td>Data Type</td><td>Signed 8-bit Integer</td><td>Unsigned 8-bit Integer</td><td>8-bit BitMap</td></tr> <tr> <td>Field Name</td><td>Average Load Adjustment Percentage Applied (O)</td><td>Duty Cycle Applied (O)</td><td>Event Control (M)</td></tr> </tbody> </table>		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)									
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)																				
	Figure D-178– Report Event Status Command Payload																						
12832	D.12.2.5.1.2 Payload Details																						
12833	The payload shall be the same as the DRLC <i>Report Event Status</i> command (see D.2.3.3.1) with the following notes:																						
12834																							
12835	Issuer Event ID (mandatory): Event ID specified in the <i>Manage Event</i> command or, if the command specified the current event, then the current running DRLC <i>Issuer Event ID</i> .																						
12836																							
12837	Event Status (mandatory): If the <i>Manage Event</i> command did not change the event, then this field will contain the last sent <i>Event Status</i> for the event. If the <i>Manage Event</i> command did change the DRLC event, and a new DRLC <i>Report Event Status</i> message will be created, then this																						
12838																							
12839																							

12840 field will contain the same value as the *Event Status* field of that message. If the command was to
12841 opt out of a critical event, a status of 0xF6 (Invalid Opt-out) will be returned. If the event ID does
12842 not exist, then a status of 0xF7 (Event Not Found) is returned. If the command is invalid, an APS
12843 default response would be returned.

12844 **Event Status Time (mandatory):** UTC Timestamp representing when the event status was
12845 changed. If the device does not know the current time but still remembers events, it may return the
12846 value of 0x00000000.

12847 D.12.2.5.1.3 When Generated

12848 This command is generated in response to a *Manage Event* Command.
12849

12850 **D.12.3 Client**

12851 **D.12.3.1 Dependencies**

12852 The client has no dependencies.

12853 **D.12.3.2 Attributes**

12854 The client has no attributes.

12855 **D.12.3.3 Commands Received**

12856 The client receives the cluster specific commands detailed in D.12.2.5.

12857 **D.12.3.4 Commands Generated**

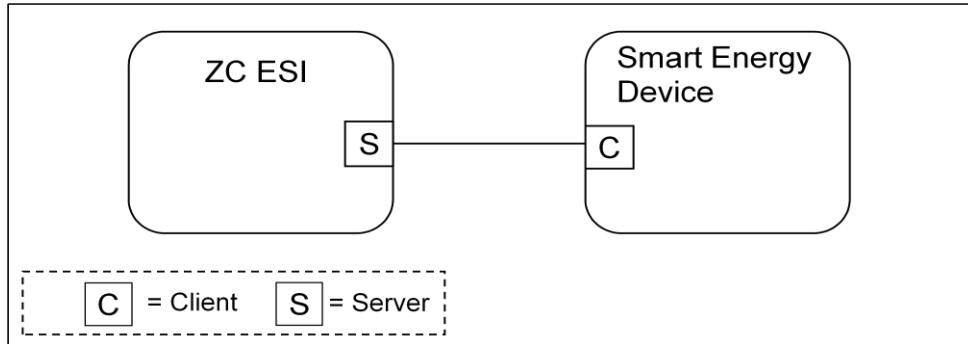
12858 The client sends the cluster specific commands detailed in D.12.2.4.

12859

12860

12861 **D.13 MDU Pairing Cluster**12862 **D.13.1 Overview**

12863 When operating within a multi-dwelling unit (MDU), the commands within this cluster allow
 12864 devices joining the NAN to acquire a list of the devices forming the ‘virtual HAN’ for the
 12865 respective household. For details on the usage of this cluster, refer to section 5, specifically
 12866 section 5.6.

12867
12868 **Figure D-179– MDU Pairing Cluster**
1286912870 **D.13.2 Server**12871 **D.13.2.1 Dependencies**

12872 Support for ZCL Data Types.

12873 **D.13.2.2 Attributes**

12874 There are no attributes for the MDU Pairing cluster server.

12875 **D.13.2.3 Commands Generated**

12876 The command IDs generated by the MDU Pairing cluster server are listed in Table D-197.

12877 **Table D-197– MDU Pairing Server Commands**

Command Identifier Field Value	Description	Mandatory / Optional
0x00	Pairing Response	O

12878

12879 D.13.2.3.1 **Pairing Response Command**

12880 The *Pairing Response* command provides a device joining a MDU network with a list of the
 12881 devices that will constitute the ‘virtual HAN’ for the household in which the joining device is to
 12882 operate.

12883 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)

12884 **Figure D-180– Pairing Response Command Payload**12885 D.13.2.3.1.2 **Payload Details**

12886 **Pairing Information Version (mandatory):** Identifies the version of pairing information
 12887 included in this command. If multiple commands are used to return this information, all
 12888 commands shall contain the same value for this field. The same version shall be used to respond
 12889 to all devices on the same ‘virtual HAN’. Newer versions replacing older versions for the same
 12890 requesting device shall use a higher value. A version of zero is an invalid value for this field.

12891 **Total Number of Devices (mandatory):** The total number of devices expected to form the
 12892 ‘virtual HAN’ (including the device to which this command is being sent). If multiple commands
 12893 are used to return this information, all commands shall contain the same value for this field.

12894 **Command Index:** The *CommandIndex* is uses to count the payload fragments in the case where
 12895 the entire payload does not fit into one message. The *CommandIndex* starts at 0 and is
 12896 incremented for each fragment belonging to the same command.

12897 **Total Number of Commands (mandatory):** In the case where the entire payload does not fit
 12898 into one message, the *Total Number of Commands* field indicates the total number of sub-
 12899 commands in the message.

12900 **EUI64 of Devices:** *EUI64 of Device 1* to *EUI64 of Device N* represent the MAC address of
 12901 devices that belong to the ‘virtual HAN’ of the requesting device; these include the requesting
 12902 device itself and all other devices the requesting device shall perform service discovery and
 12903 binding with. Should one message be insufficient to transfer all EUI64s, additional messages will
 12904 be required (see *Command Index* and *Total Number of Commands* fields).

12905 D.13.2.3.1.3 When Generated

12906 This command is generated by a TC ESI controlling a MDU network when a *Pairing Request*
 12907 command is received (see D.13.3.3.1). Receipt of a *Pairing Request* command by a TC ESI that
 12908 is not operating in a MDU, or receipt by an ESI that is not the ZC/TC, shall result in a ZCL
 12909 Default Response being returned with a status of UNSUP_CLUSTER_COMMAND.

12910 D.13.2.3.1.4 Effect on Receipt

12911 On receipt of this command, the requesting device shall limit the devices with which it may
 12912 perform service discovery and binding to those identified in the response, instead of all devices
 12913 on the ZigBee network to which it has joined. Where the entire payload of the response will not
 12914 fit into a single command, the receiving device shall ensure that it successfully receives all
 12915 payload fragments before the payload is used.

12916

12917 **D.13.3 Client**

12918 **D.13.3.1 Dependencies**

12919 Support for ZCL Data Types.

12920 **D.13.3.2 Attributes**

12921 There are no attributes for the MDU Pairing cluster client.

12922 **D.13.3.3 Commands Generated**

12923 The command IDs generated by the MDU Pairing cluster client are listed in Table D-198.

12924

Table D-198– MDU Pairing Client Commands

Command Identifier Field Value	Description	Mandatory / Optional
0x00	Pairing Request	M

12925

12926 **D.13.3.3.1 Pairing Request Command**

12927 The *Pairing Request* command allows a device joining a MDU network to determine the devices
 12928 that will constitute the ‘virtual HAN’ for the household in which it is to operate.

12929 **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-181—Pairing Request Command Payload

12930

12931 D.13.3.3.1.2 Payload Details

12932 **Local Pairing Information Version (mandatory):** Identifies the version of pairing information
 12933 currently held on the requesting device. The TC ESI shall use this information to determine if
 12934 new pairing information is to be returned. A device that does not yet have local pairing
 12935 information shall set this field to zero (0x00000000).

12936 **EUI64 of Requesting Device (mandatory):** Identifies the MAC address of the requesting
 12937 device.

12938 D.13.3.3.1.3 When Generated

12939 A device should send this command to the TC ESI of a network that it is in the process of
 12940 joining. It should also periodically send the command to the TC ESI in order to determine if
 12941 further devices have joined its ‘virtual HAN’ or that the devices constituting the ‘virtual HAN’
 12942 have changed (*Pairing Information Version* has been updated).

12943 D.13.3.3.1.4 Effect on Receipt

12944 If the command is received by a TC ESI that is not operating in a MDU, or the receiving ESI is
 12945 not the TC ESI, a ZCL Default Response shall be returned with a status of
 12946 UNSUP_CLUSTER_COMMAND (the joining device should then follow normal (traditional)
 12947 service discovery and binding procedures).

12948 If the command contains a *Local Pairing Information Version* that matches the version of pairing
 12949 information for the requesting device that is held by the TC ESI, a ZCL Default Response shall
 12950 be returned with a status of WAIT_FOR_DATA (the same response should be returned if data is
 12951 not yet available on the TC ESI for the requesting device).

12952 If the command is received by a TC ESI that is controlling a MDU, the TC ESI should return a
Pairing Response command indicating the devices that will constitute the ‘virtual HAN’ for the
 12953 household in which the requesting device is to operate. A joining device should then follow
 12954 normal (traditional) service discovery and binding procedures, but only with those devices
 12955 identified in the response command.

12957 Note that this command should not be recognized by an older Smart Energy TC ESI; this should
 12958 result in that TC ESI returning a ZCL Default Response with a status of
 12959 UNSUP_CLUSTER_COMMAND.

12960

Annex E RULES AND GUIDELINES FOR OVERLAPPING EVENTS

This section describes multiple scenarios that Demand Response and Load Control devices may encounter over the Smart Energy network. The examples describe situations of overlapping events that are acceptable and where overlapping events that will be superseded due to conflicts.

E.1 Definitions

Start Time – “Start Time” field contained within the Load Control Event packet indicating when the event should start. Please note, a “Start Time” value of 0x00000000 denotes “now” and the device should use its current time as the “Start Time”.

Duration – “Duration” field contained within the Load Control Event packet indicating how long the event should occur.

End Time – Time when Event completes as calculated by adding *Duration* to *Start Time*.

Scheduled Period - Represents the time between the *Start Time* and the *End Time* of the event.

Effective Start Time - Represents time at which a specific device starts a load control event based on the *Start Time* plus or minus any randomization offsets.

Effective End Time - Represents time at which a specific device ends a load control event based on the *Start Time* plus *Duration*, plus or minus any randomization offsets.

Effective Scheduled Period - Represents the time between the *Effective Start Time* and the *Effective End Time*.

Overlapping Event - Defined as an event where the *Scheduled Period* covers part or all of an existing, previously scheduled event.

Successive Events - Defined as two events where the scheduled *End Time* of the first event is equal the *Start Time* of a subsequent scheduled event.

Nested Events - Defined as two events where the scheduled *Start Time* and *End Time* of the second event falls during the *Scheduled Period* of the first scheduled event and the second event is of shorter duration than the first event.

12988 **E.2 Rules and Guideline**

12989 The depicted behaviors and required application management decisions are driven from the
12990 following guidance and rule set:

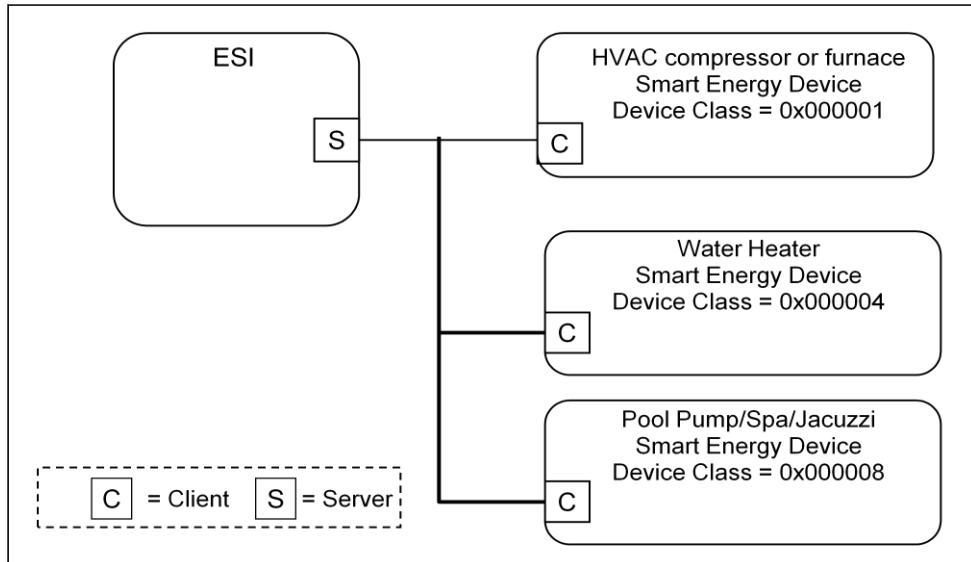
- 12991 **1** Upstream Demand Response/Load Control systems and/or the ESI shall prevent
12992 mismanaged scheduling of *Overlapping Events* or *Nested Events*. It is recognized Upstream
12993 Demand Response/Load Control systems and/or the ESI will need to react to changing
12994 conditions on the grid by sending *Overlapping Events* or *Nested Events* to supersede previous
12995 directives. But those systems must have the proper auditing and management rules to prevent
12996 a cascading set of error conditions propagated by improperly scheduled events.
- 12997 **2** When needed, Upstream Demand Response/Load Control systems and/or the ESI may
12998 resolve any event scheduling conflicts by performing one of the following processes:
12999 a Canceling individual events starting with the earliest scheduled event and re- issuing a new
13000 set of events.
13001 b Canceling all scheduled events and re-issuing a new set of events.
13002 c Sending *Overlapping Events* or *Nested Events* to supersede previous directives.
13003 It is recommended that process 2.c is used for most situations since it can allow a smoother
13004 change between two sets of directives, but no way does it negate the responsibilities identified
13005 in rule #1.
- 13006 **3** When an End Device receives an event with the *End Time* in the past (*End Time* < Current
13007 Time), this event is ignored and a *Report Event Status* command is returned with the Event
13008 Status set to 0xFB (Rejected - Event was received after it had expired).
- 13009 **4** When an End Device receives an event with a *Start Time* in the past and an *End Time* in the
13010 future ((*Start Time* < Current Time) AND (*End Time* > Current Time)), the event is processed
13011 immediately. The Effective *Start Time* is calculated using the Current Time as the *Start Time*.
13012 Original *End Time* is preserved.
- 13013 **5** Regardless of the state of an event (scheduled or executing), when an *End Device* detects an
13014 *Overlapping Event* condition the latest *Overlapping Event* will take precedence over the
13015 previous event. Depending on the state of the event (scheduled or executing), one of the
13016 following steps shall take place:
13017 a If the previous event is scheduled and not executing, the End Device returns a *Report*
13018 *Event Status* command (referencing the previous event) with the Event Status set to 0x07
13019 (The event has been superseded). After the *Report Event Status* command is successfully
13020 sent, the End Device can remove the previous event schedule.

- 13021 b If the previous event is executing, the End Device shall change directly from its current
13022 state to the requested state at the *Effective Start Time* of the *Overlapping Event* (Note:
13023 Rule #4 effects *Effective Start Time*). The End Device returns a *Report Event Status*
13024 command (referencing the previous event) with the Event Status set to 0x07 (the event has
13025 been superseded).
- 13026 6 Randomization **shall not** cause event conflicts or unmanaged gaps. To clarify:
- 13027 a When event starting randomization is requested, time periods between the *Start Time* of an
13028 event and the *Effective Start Time* a device should either maintain its current state or apply
13029 changes which contribute to energy saving. Preference would be to maintain current state.
- 13030 b When event ending randomization is used and the *Effective End Time* overlaps the
13031 *Effective Start Time* of a *Successive Event*, the *Effective Start Time* takes precedence.
13032 Events are not reported as superseded, End devices should report event status as it would a
13033 normal set of *Successive Events*.
- 13034 c It is recommended devices apply the same Start and Stop Randomization values for
13035 consecutive events to help prevent unexpected gaps between events.
- 13036 d Devices **shall not** artificially create a gap between *Successive Events*.
- 13037 7 It is permissible to have gaps when events are not *Successive Events* or *Overlapping Events*.
- 13038 8 If multiple device classes are identified for an event, future events for individual device
13039 classes (or a subset of the original event) that cause an *Overlapping Event* will supersede the
13040 original event strictly for that device class (or a subset of the original event). Note: Rule #5
13041 applies to all *Overlapping Events*.

13042

13043 **E.3 Event Examples**

13044 Smart Energy devices which act upon Demand Response and Load Control events shall use the
13045 following examples for understanding and managing overlapping and superseded events.
13046 Within those examples, references to multiple device classes will be used. Figure E-1 depicts
13047 a representation of those devices in a Smart Energy network.



13048

13049

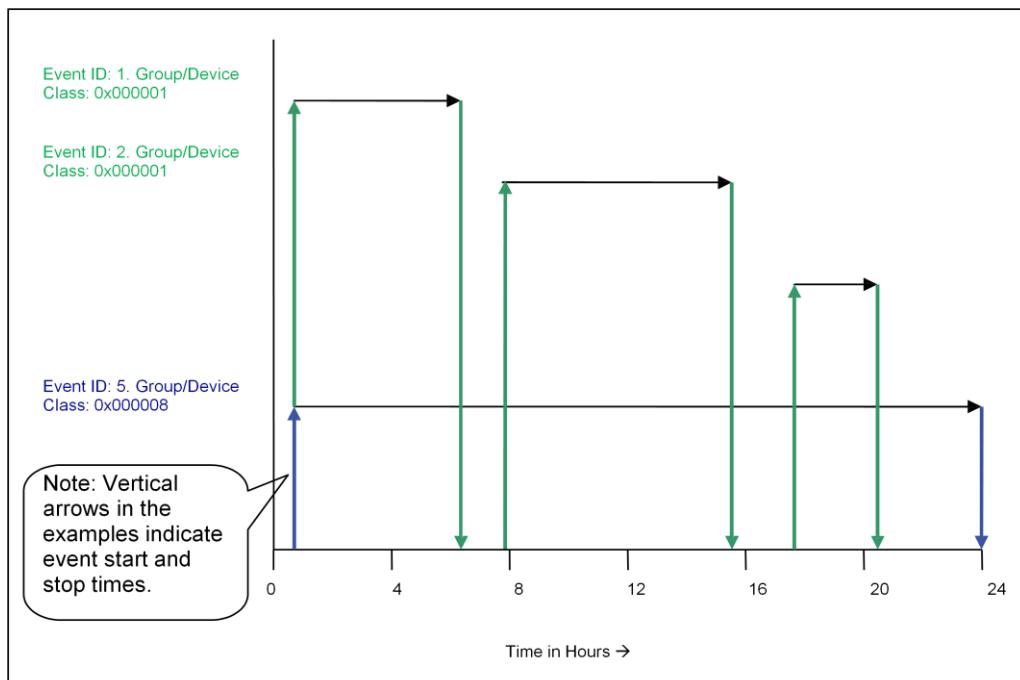
Figure E-1– Smart Energy Device Class Reference Example

13050

E.3.1 Correct Overlapping Events for Different Device Classes

13051

Figure E-2 depicts a correct series of DR/LC event for device class of 0x000001 (reference for the BitMap definition) with an event scheduled for another device class during the same period.



13053

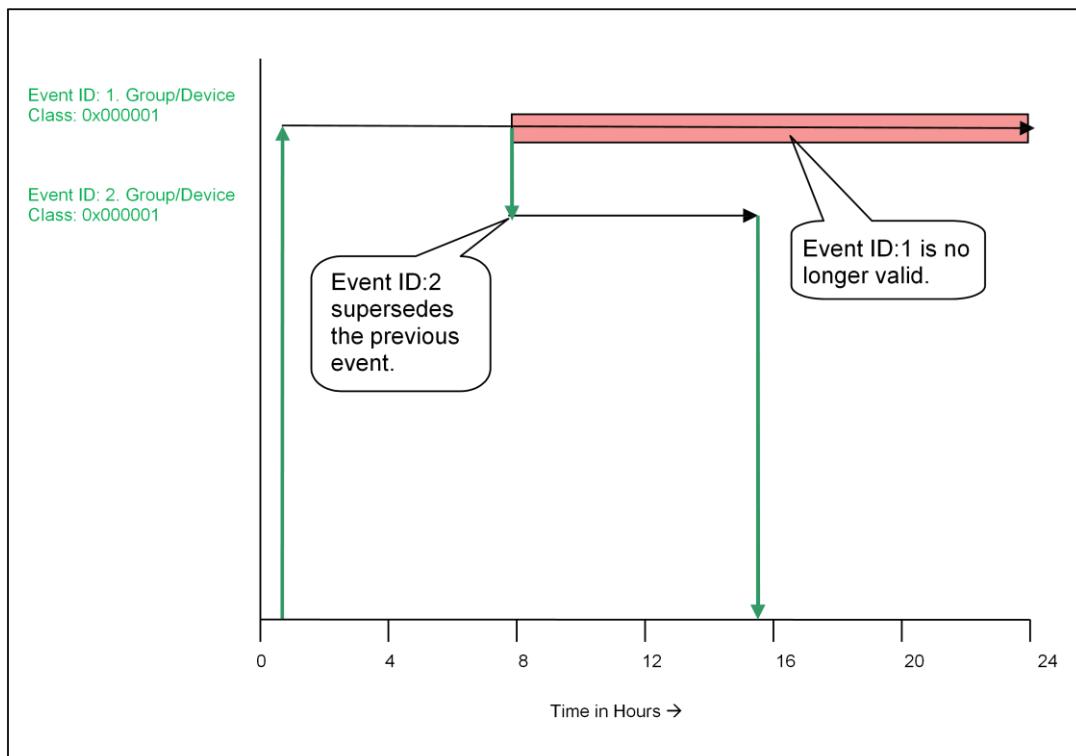
13054

Figure E-2– Correctly Overlapping Events

13055 In Figure E-2, Device Class 0x000001 receives a sequence of 3 unique DR/LC events to be
 13056 scheduled and acted upon. During this same 24 hour period, Device Class 0x000008 receives
 13057 one scheduled DR/LC event that spans across the same time period as the events scheduled for
 13058 Device Class 0x0000001. Because both Device Classes are unique, there are no conflicts due to
 13059 Overlapping Events.

13060 **E.3.2 Correct Superseded Event for a Device Class**

13061 Figure E-3 below depicts a correct series of DR/LC events for device class of 0x000001
 13062 (reference for the BitMap definition) where an event is scheduled then later superseded.



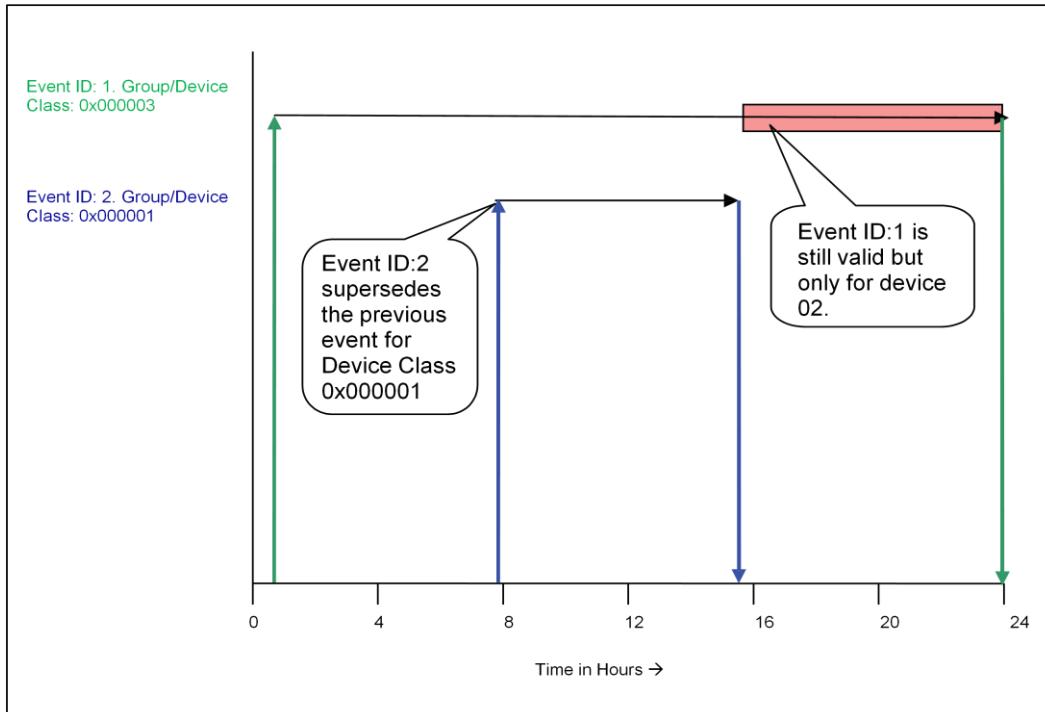
13063

Figure E-3– Correct Superseding of Events

13065 In Figure E-3, Device Class 0x000001 receives DR/LC Event ID#1 setup for a 24 hour
 13066 *Scheduled Period*, which later is superseded by DR/LC Event ID#2, invalidating the
 13067 remainder of Event ID#1, which is cancelled.

13068 **E.3.3 Superseding Events for Subsets of Device Classes**

13069 Figure E-4 below depicts a correct series of DR/LC events for device class of 0x000001
 13070 (reference for the BitMap definition) with an event scheduled for another device class during
 13071 the same time period.



13072

13073

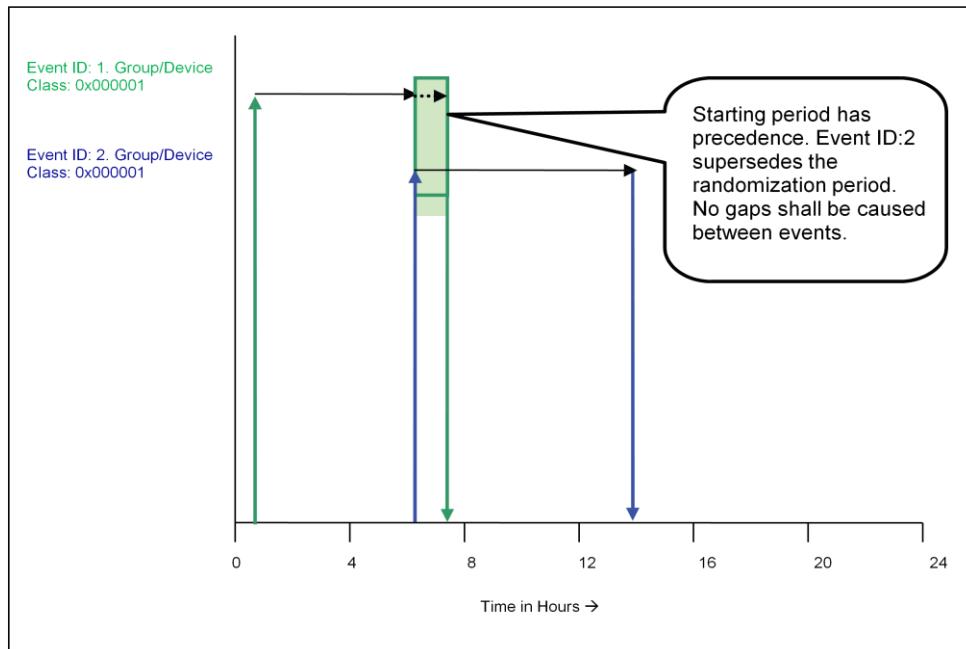
Figure E-4– Superseded Event for a Subset of Device Classes

13074 In Figure E-4, Device Class 0x000003 receives DR/LC Event ID#1 setup for a 24 hour
 13075 *Scheduled Period*, which is targeted for both Device Class 0x000002 and 0x000001 (OR'ed
 13076 == 0x000003). In the example, Event ID#2 is issued only for Device Class 0x000001,
 13077 invalidating the remainder of Event ID#1 for that device class. DR/LC Event ID#1 is still valid
 13078 for Device Class 0x000002, which in the example should run to completion.

13079 **E.3.4 Ending Randomization Between Events**

13080 Figure E-5 below depicts an *Effective End Time* that overlaps a second scheduled DR/LC event
 13081 for device class of 0x000001 (reference for the BitMap definition).

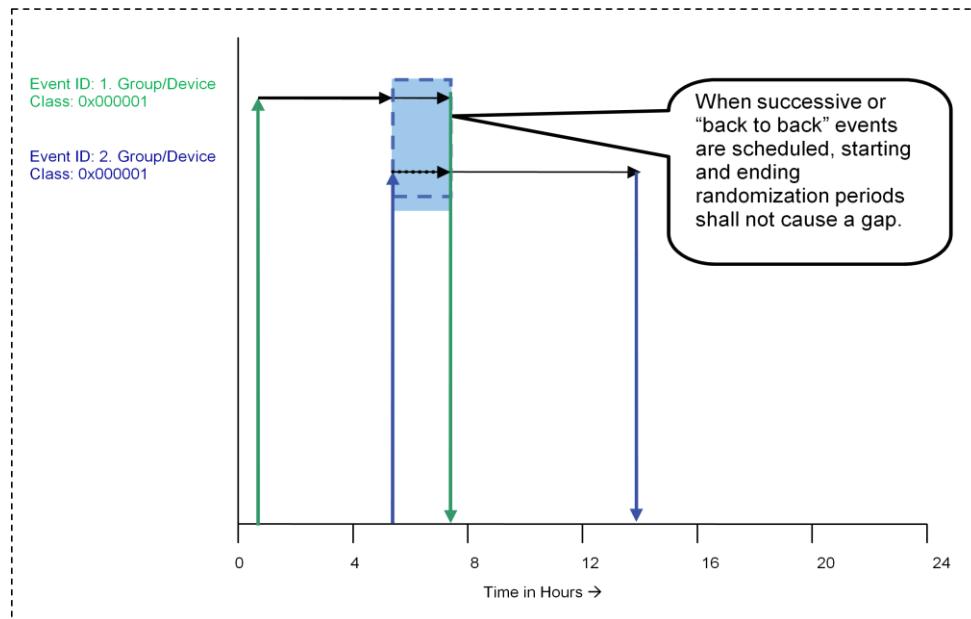
13082

13083
13084**Figure E-5– Ending Randomization Between Events**

13085 In Figure E-5, Device Class 0x000001 receives a DR/LC Event ID#1 with an ending
 13086 randomization setting (please refer to sub-clause D.2.2.3.1.1.1 for more detail). A second
 13087 DR/LC (Event ID#2) is issued with a starting time which matches the ending time of DR/LC
 13088 Event ID#1. In this situation, the *Start Time* of Event ID#2 has precedence. Event ID#1 is not
 13089 reported as superseded.

13090 **E.3.5 Start Randomization Between Events**

13091 Figure E-6 below depicts an *Effective Start Time* that overlaps a previously scheduled
 13092 DR/LC event for device class of 0x000001 (reference for the BitMap definition).

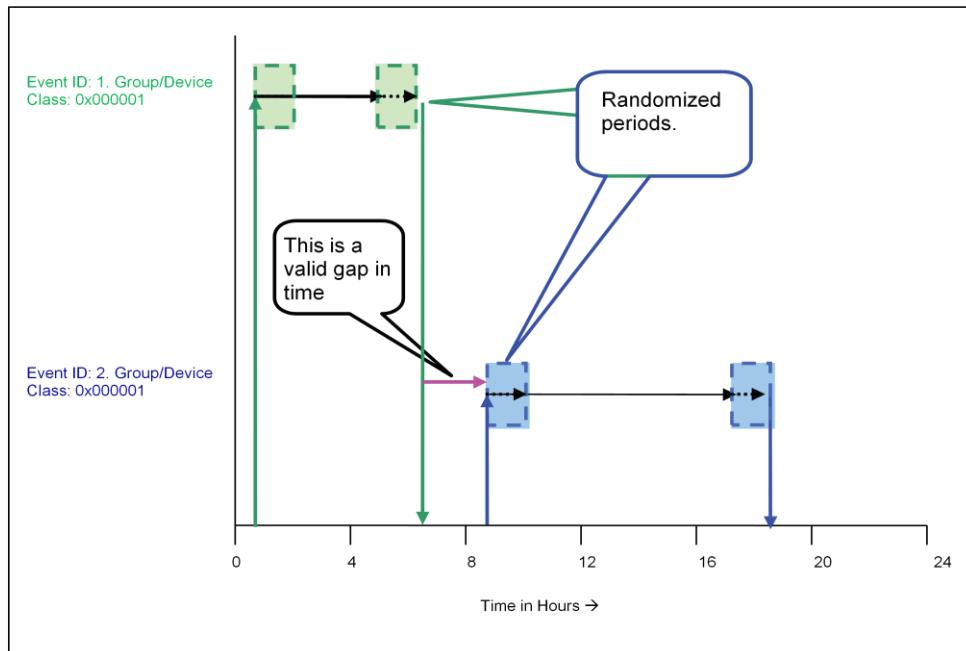
13093
13094**Figure E-6– Start Randomization Between Events**

13095 Figure E-6 above, Device Class 0x000001 receives a DR/LC Event ID#1 with an ending
 13096 randomization setting (please refer to sub-clause D.2.2.3.1.1 for more detail). *Effective End*
 13097 *Time* of Event ID#1 is not known. A second DR/LC (Event ID#2) is issued with a starting
 13098 randomized setting, which has an *Effective Start Time* that could overlap or start after the
 13099 *Effective End Time* of DR/LC Event ID#1. In this situation, the *Effective Start Time* of Event
 13100 ID#2 has precedence but the DR/LC device must also prevent any artificial gaps caused by the
 13101 *Effective Start Time* of Event ID#2 and *Effective End Time* of Event ID#1.

13102

13103 **E.3.6 Acceptable Gaps Caused by Start and Stop Randomization of Events**

13104 Figure E-7 below depicts an acceptable gap between two scheduled DR/LC events for device
 13105 class of 0x000001 (reference for the BitMap definition) using both starting and ending
 13106 randomization with both events.



13107
13108

Figure E-7– Acceptable Gaps with Start and Stop Randomization

13109 Figure E-7 above, Device Class 0x000001 receives a DR/LC Event ID#1 with both a starting
13110 and ending randomization setting (please refer to sub-clause D.2.2.3.1.1.1 for more detail). A
13111 second DR/LC Event ID#2 is also issued with both a starting and ending randomized setting.
13112 The primary configuration to note in this example is the *Effective End Time* of DR/LC Event
13113 ID#1 completes well in advance of the *Effective Start Time* of DR/LC Event ID#2. In this
13114 scenario, regardless of randomization a gap is naturally created by the scheduling of the events
13115 and is acceptable.

13116

Annex F JOINING PROCEDURE USING PRE-CONFIGURED TRUST CENTER LINK KEYS

13117
13118
13119

13120 The secure join procedure is detailed as follows:

- 13121 • The secured joining procedure is as stated in [B3] Section 4.6.3.2.3. The case used in the
13122 Smart Energy application is the “Pre-configured trust center link key and address”
- 13123 • In [B3] Section 4.6.3.2.3.2, in the case of “Pre-configured Trust Center Link Key”, the
13124 joining device waits for the APSME-TRANSPORT-KEY.Indication. The frame is
13125 encrypted/authenticated with the key-transport key according to the methodologies specified
13126 in sections 4.4.1.1 and 4.5.3 of the ZigBee specification r17, which describe the key-
13127 transport keys and their association with link keys, in this case the pre-configured trust center
13128 link key. The source address will be that of the Trust Center. The key transported will be the
13129 NWK Key Key type == 0x01.
- 13130 • When the trust center sends the tunneled *Transport Key* command, the Extended Nonce bit
13131 on the Auxiliary Frame Header must be set to 1 on the Transport Key frame from the Trust
13132 Center to the joining child as described in [B3] Section 4.5.1. The Trust Center must also
13133 insert its long address into the Source Address field of the Auxiliary Frame Header since that
13134 information will be needed at the child to decrypt the *Transport Key* command.
- 13135 • Sub-clause 5.4 of this document calls out two cases for secured join: pre- configured link
13136 keys and temporary link keys. The joining device and trust center perform the same join
13137 operation in both cases. The only difference is how the joining device and trust center treat
13138 the initial key material (either using it directly as the pre-configured link key or hashing with
13139 some data like the long address of the joining device at application level first, see Annex E
13140 for this method). From the perspective of the security joining process what happens
13141 afterwards is the secure join procedure is the same.
- 13142 • In either case called out in sub-clause 5.4 of this document, the joining device is
13143 authenticated using the [B3] Section 4.6.3.2.3.2 procedure or leaves if the security timeout
13144 expires. If authenticated, the key delivered via the APSME- TRANSPORT-KEY.indication
13145 in [B3] Section 4.6.3.2.3.2 is the same for either case called out in the AMI specification sub-
13146 clause 5.4 (no matter how the application determined the pre-configured link key).

13147 In terms of the message exchange between the child and trust center in performing the secure join
13148 procedure, the following is employed:

- 13149 1 Child joining device uses NLME-JOIN.request to parent. Parent sends an APSME-
13150 UPDATE-DEVICE.request to the Trust Center on behalf of the child to the Trust Center.

- 13151 APSME-UPDATE-DEVICE.request is transported encrypted/authenticated with the NWK
13152 key that the parent has
- 13153 **2** Upon receipt at the trust center, the trust center must perform the following processing:
- 13154 **a** Validity check of the child's address to determine if a trust center link key exists between
13155 the trust center and the address provided by the joining child.
- 13156 **b** If the child has the trust center as its parent, the APSME-TRANSPORT- KEY.request is
13157 sent directly to the child encrypted with the key-transport key derived from the trust center
13158 link key known to the child device and the trust center, ELSE
- 13159 *i.* If the child does not have the trust center as its parent, the APSME- TRANSPORT-
13160 KEY command frame is encrypted using the key-transport key derived from the
13161 trust center link key shared between the child and the trust center.
- 13162 **c** The resulting encrypted payload is sent to the child using the *APS Tunnel* command. The
13163 *APS Tunnel* command and its (already encrypted) payload is encrypted using the NWK
13164 key from the trust center to the child's parent. On the final hop, the child's parent will
13165 perform the following processing according to [B3] Section 4.6.3.7.2:
- 13166 *i.* The parent sends the contents within the *APS Tunnel* command to the child without
13167 network layer encryption. The message from the parent to the joining child is an APS
13168 encrypted transport key command using the key-transport key derived from the trust
13169 center link key.
- 13170 Here are the details on the message that is routed from the trust center to the joining device's
13171 parent via the *Tunnel* command:
- 13172
 - NWK Data Frame (Dest: Parent)
 - APS Header (Command)
 - APS Command Frame (Tunnel)
 - Dest EUI: Child
 - Tunnel Payload
 - APS Header
 - APS Auxiliary Header
 - Encrypted Payload
 - APS Command Frame (Transport Key)

13181 Here are the details on the message that is routed from the joining device's parent to the joining
13182 child:

- 13183 • NWK Data Frame (added by parent, Dest: child)
- 13184 • APS Header (from Tunnel Payload)
- 13185 • APS Auxiliary Header (from Tunnel Payload)
- 13186 • Encrypted Payload (from Tunnel Payload)
- 13187 • APS Command Frame (Transport Key)

13188 The message to the child from the parent is identical if the device joins directly to the Trust
13189 Center.

13190 As a note on the final hop contents of the payload:

- 13191 • The last hop of the APME-TRANSPORT-KEY message from parent to joining child has NO
13192 network layer encryption, but does have application layer encryption
- 13193 • Thus: There will be no NWK auxiliary header, but there will be an APS auxiliary header
- 13194 • The APS auxiliary header will have the Key Identifier Sub-Field set to 0x02 == A key-
13195 transport key (see [B3] Section 4.5.1.1.2)
- 13196 • The APS frame will be encrypted with the key-transport key derived from the pre-configured
13197 trust center link key. The pre-configured trust center link key must be part of the
13198 apsDeviceKeyPairSet in the AIB of the joining device and also known to the trust center.
- 13199 • The resulting APS frame from the parent to the joining child is the APS- TRANSPORT-KEY
13200 message encrypted with the key-transport key derived from the trust center link key delivered
13201 with the key type of key-transport key (0x02).
- 13202 • Per [B3] Section 4.4.3.2, the KeyType field will be set to (0x01) == Network Key
- 13203 • The TransportKeyData will be the active network key and sequence number
- 13204 • The joining device must set the network key and sequence number in its NWK Information
13205 Block.
- 13206 • The device is then joined and authenticated.

13207
13208

13209
13210
13211
13212
13213
13214
13215
13216
13217
13218
13219
13220

--- END OF DOCUMENT ---