Abstract:
This document defines the advanced metering initiative profile.

Keywords:
ZigBee, Profile, Advanced Metering Initiative, Application Framework.
Contact information

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

Phil Jamieson,
Philips, Cross Oak Lane,
Redhill, Surrey, RH1 5HA, UK
E-Mail: phil.jamieson@philips.com
Phone: +44 1293 815265
Fax: +44 1293 815050

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

The information on this page should be removed when this document is accepted.
The following is a list of those who were members of the ZigBee Alliance Application Framework Working Group leadership when this document was released:

- **Phil Jamieson**: Chair
- **Don Sturek**: Editor-in-chief
- **Drew Gislason**: Secretary

When the document was released, the Advanced Metering Initiative Profile Task Group leadership was composed of the following members:

- **Robby Simpson**: Chair
- **Tim Gillman**: Vice chair
- **John Buffington**: Technical Editor
- **Matt Maupin**: Program Manager

Contributions were made to this document from the following members:

- Mads Westergreen
- Jeff Huggins
- Gary Birk
- Don Sturek
- Skip Aston
- Robert Cragie
- Zachary Smith
- Howard Ng
- Michel Veillette
- John Cowburn
- Damon Corbin
- Tim Enwall
- Rick Bojahra
- Juan Agüí Martín
- Jakob Thomsen
- Michael G. Stuber
- Jeff Mathews
- John Prater
### Table of Contents

1. Introduction .................................................................................................................. 1
   1.1 Scope .......................................................................................................................... 1
   1.2 Purpose ....................................................................................................................... 1

2. References .................................................................................................................... 2
   2.1 ZigBee Alliance documents ....................................................................................... 2
      2.1.1 External Reference documents ........................................................................ 2

3. Definitions .................................................................................................................... 4
   3.1 Conformance levels .................................................................................................... 4
   3.2 ZigBee Definitions ...................................................................................................... 4

4. Acronyms and abbreviations ......................................................................................... 5

5. Profile description ......................................................................................................... 7
   5.1 A ZigBee AMI Network ............................................................................................. 7
   5.2 ZigBee stack profile .................................................................................................. 9
      5.2.1 ZigBee Routing Table Size Recommendations ................................................. 9
      5.2.2 ZigBee Coordinator and Trust Center Recommendations ................................ 10
   5.3 Startup Attribute Set (SAS) .................................................................................... 10
      5.3.1 Start Up Parameters ......................................................................................... 10
      5.3.2 Join Parameters ............................................................................................... 11
      5.3.3 Security Parameters ........................................................................................ 11
      5.3.4 End Device Parameters .................................................................................... 11
      5.3.5 Link Status Parameters ................................................................................... 11
      5.3.6 Concentrator Parameters ............................................................................... 11
      5.3.7 APS Transport Parameters ............................................................................. 12
      5.3.8 Binding Parameters ........................................................................................ 12
   5.4 AMI Security ............................................................................................................. 12
      5.4.1 Joining with Preinstalled Link Keys ................................................................ 12
      5.4.2 Joining using the Key Establishment Cluster ................................................ 13
   5.5 Public Pricing ........................................................................................................... 13

6. Other AMI Requirements and Best Practices .............................................................. 13
   5.6 Coexistence and Interoperability With HA Devices ............................................... 14
   5.7 Device descriptions ................................................................................................ 14
   5.8 ZigBee Cluster Library (ZCL) .................................................................................. 15
   5.9 Cluster List ............................................................................................................... 15
      5.9.1 General Clusters ............................................................................................ 16
   5.10 AMI Cluster Descriptions ...................................................................................... 17
      5.10.1 Demand Response and Load Control Cluster ................................................ 17
      5.10.2 Simple Metering Cluster ............................................................................... 32
      5.10.3 Price Cluster ................................................................................................ 42
      5.10.4 Messaging Cluster ......................................................................................... 46
      5.10.5 Registration Cluster ...................................................................................... 51
   5.11 Commissioning ........................................................................................................ 54
      5.11.1 Forming the network (Start-up sequence) ...................................................... 55
      5.11.2 Support for commissioning modes .................................................................. 55
   5.12 Commissioning Documentation ............................................................................ 55
   5.13 Commissioning Procedure for Different Network Types ..................................... 56
   5.14 AMI Tunneling (Complex Metering) Cluster ......................................................... 59
   5.15 Pre-Payment Cluster .............................................................................................. 59

7. Constants, error codes and general alarms ................................................................... 60

8. Device Specifications .................................................................................................... 61

---

This is an unaccepted ZigBee specification draft, subject to change.
| 1 | C.2.1.2.1 Dependencies ......................................................... | 91 |
| 2 | C.2.1.2.2 Attributes ......................................................... | 91 |
| 3 | C.2.1.2.3 Information ......................................................... | 92 |
| 4 | C.2.1.2.3.1 SecuritySuite attribute ....................................... | 92 |
| 5 | C.2.1.2.4 Settings .............................................................. | 92 |
| 6 | C.2.1.2.5 Commands Received ................................................ | 92 |
| 7 | C.2.1.2.5.1 Initiate Key Establishment command.......................... | 95 |
| 8 | C.2.1.2.5.1.1 Payload Format ................................................ | 95 |
| 9 | C.2.1.2.5.1.2 Effect on Receipt ............................................. | 95 |
| 10 | C.2.1.2.5.2 Confirm Key command ........................................... | 95 |
| 11 | C.2.1.2.5.2.1 Payload Format ................................................ | 95 |
| 12 | C.2.1.2.5.2.2 Effect on Receipt ............................................. | 95 |
| 13 | C.2.1.2.5.3 Terminate Key Establishment command........................ | 95 |
| 14 | C.2.1.2.5.3.1 Payload Format ................................................ | 95 |
| 15 | C.2.1.2.5.3.2 Effect on Receipt ............................................. | 95 |
| 16 | C.2.1.2.6 Commands Generated ................................................ | 95 |
| 17 | C.2.1.2.6.1 Initiate Key Establishment response command ............... | 95 |
| 18 | C.2.1.2.6.1.1 Payload Format ................................................ | 95 |
| 19 | C.2.1.2.6.2 Confirm Key response command .................................. | 95 |
| 20 | C.2.1.2.6.2.1 Payload Format ................................................ | 95 |
| 21 | C.2.1.3 Client ................................................................. | 95 |
| 22 | C.2.1.3.1 Dependencies ....................................................... | 96 |
| 23 | C.2.1.3.2 Attributes ........................................................... | 96 |
| 24 | C.2.1.3.3 Commands Received ................................................ | 96 |
| 25 | C.2.1.3.4 Commands Generated ................................................ | 96 |
| 26 | C.3 Application implementation ............................................... | 96 |
| 27 | C.3.1 Network security for AMI networks .................................... | 96 |
| 28 | C.3.2 Certificate-Based Key Establishment ................................ | 97 |
| 29 | C.3.2.1 Notation and representation ....................................... | 98 |
| 30 | C.3.2.1.1 Strings and string operations ................................... | 98 |
| 31 | C.3.2.1.2 Integers and their representation ................................ | 98 |
| 32 | C.3.2.1.3 Entities ............................................................. | 98 |
| 33 | C.3.2.2 Elliptic curve domain parameters .................................. | 98 |
| 34 | C.3.2.2.2 Elliptic-curve point representation ................................ | 98 |
| 35 | C.3.2.2.3 Elliptic-curve public-key pair ................................... | 98 |
| 36 | C.3.2.2.4 ECC implicit certificates ....................................... | 99 |
| 37 | C.3.2.2.5 ECC Block-cipher .................................................. | 99 |
| 38 | C.3.2.2.6 ECC Cryptographic Hash Function ................................ | 99 |
| 39 | C.3.2.2.7 ECC Keyed hash function for message authentication .......... | 99 |
| 40 | C.3.2.3 CBKE key agreement scheme ........................................ | 100 |
| 41 | C.3.3 Certificate-Based Key Establishment ................................ | 100 |
| 42 | C.3.3.1 Exchange Ephemeral Data ............................................ | 101 |
| 43 | C.3.3.1.1 Initiator ............................................................. | 101 |
| 44 | C.3.3.1.2 Responder ........................................................... | 101 |
| 45 | C.3.3.2 Validate Implicit Certificates ..................................... | 102 |
| 46 | C.3.3.2.1 Initiator ............................................................. | 102 |
| 47 | C.3.3.2.2 Responder ........................................................... | 102 |
| 48 | C.3.3.3 Derive Keying Material ............................................. | 102 |
| 49 | C.3.3.3.1 Initiator ............................................................. | 102 |
| 50 | C.3.3.3.2 Responder ........................................................... | 102 |
| 51 | C.3.3.4 Confirm Keys .......................................................... | 102 |
| 52 | C.3.3.4.1 Initiator ............................................................. | 102 |
| 53 | C.3.3.4.2 Responder ........................................................... | 102 |
## List of Figures

1. Figure 1 Utility Private HAN ................................................................. 8
2. Figure 2 Utility Private NAN ............................................................... 8
3. Figure 3 Customer Private HAN ........................................................... 9
4. Figure 4 – Format of the Load Control Event payload .................................. 18
5. Figure 5 – Format of the Cancel Load Control Event payload ............................ 22
6. Figure 6 – Format of the Cancel All Load Control Events payload ....................... 23
7. Figure 7 – Format of the Report Event Status Command payload ......................... 25
8. Figure 8 - Example of both a successful and an overridden Load Curtailment event ... 30
9. Figure 9 - Example of a Load Curtailment superceded and another cancelled ........... 32
10. Figure 10 Stand Alone ESP Model w/Mains Powered Metering Device ................. 33
11. Figure 11 Standalone ESP Model w/Battery Powered Metering Device .................. 33
12. Figure 12 ESP Model w/Integrated Metering Device ...................................... 34
13. Figure 13 Get Profile Response Command Payload ....................................... 41
14. Figure 14 The Format of the Get Current Price command payload ....................... 43
15. Figure 15 Format of the Publish Price Command payload .................................. 45
16. Figure 16 Messaging Cluster Client/Server Example ...................................... 47
17. Figure 17 Client/Server Message Command Exchanges ................................... 51
18. Figure 18 Registration Cluster Client/Server Example ...................................... 51
19. Figure 19 Successful network startup and certificate exchange ............................ 58
20. Figure 20 - ZigBee Stack with Stub APS .................................................. 76
21. Figure 21 - Normal ZigBee Frame ............................................................ 82
22. Figure 22 - Inter-PAN ZigBee Frame ........................................................ 82
23. Figure 23 - Stub NWK Header Format ....................................................... 82
24. Figure 24 – Format of the NWK Frame Control Field .................................... 83
25. Figure 25 – Stub APS Header Format ....................................................... 83
26. Figure 26 - Format of the APS Frame Control Field ....................................... 83
27. Figure 27 – Inter-PAN Typical Usage ....................................................... 86
28. Figure 28 – Overview of general exchange ................................................ 89
29. Figure 29 – Typical Usage of the key establishment cluster ................................ 90
30. Figure 30 – Format of the Initiate Key Establishment command payload .................. 93
31. Figure 31 – Format of the Confirm Key command payload ................................ 93
32. Figure 32 – Format of the Terminate Key Establishment command payload ............. 94
33. Figure 33 – Format of the Initiate Key Establishment response command payload ....... 95
34. Figure 34 – Format of the Confirm Key response command payload .................... 96
# List of Tables

1. **Table 1** – Document revision change history
2. **Table 2** – Devices specified in the AMI profile
3. **Table 3** – Clusters used in the AMI profile
4. **Table 6** – Demand Response Client Cluster Attributes
5. **Table 7** – Generated Command IDs for the Demand Response and Load Control client
6. **Table 9** – Summation Information Attribute Set
7. **Table 10** – Meter Status Attribute Set
8. **Table 11** – Consumption Formatting Attribute Set
9. **Table 12** – ESP Historical Attribute Set
10. **Table 13** – Load Profile Configuration Attribute Set
11. **Table 14** – Generated Command IDs for the Simple Metering server
12. **Table 15** – Generated Command IDs for the Messaging server
13. **Table 16** – Display Message command payload
14. **Table 17** – Cancel Message command payload
15. **Table 18** – Messaging Client Commands
16. **Table 19** – Registration Cluster Attributes
17. **Table 20** – Generated Command IDs for the Registration server Cluster
18. **Table 21** – Constants Specific to the AMI Profile
19. **Table 22** – Clusters common to all devices
20. **Table 23** – Clusters Supported by the Energy Service Portal
21. **Table 24** – Clusters Supported by the Metering Device
22. **Table 25** – Clusters Supported by the In-Premise Display Device
23. **Table 26** – Clusters Supported by the PCT
24. **Table 27** – Clusters Supported by the Load Control Device
25. **Table 28** – AMI Clusters Supported by the Range Extender Device
26. **Table 29** – Clusters Supported by the Smart Appliance Device
27. **Table 30** – Additional Time Cluster Data type
28. **Table 31** – Parameters of the INTRP-DATA.request
29. **Table 32** – Parameters of the INTRP-DATA.confirm
30. **Table 33** – Parameters of the INTRP-DATA.indication
Table 1 shows the change history for this specification.

<table>
<thead>
<tr>
<th>Revision</th>
<th>Version</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>Original version.</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>First draft to include annexes and cluster information.</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>Updated to include Key Establishment Cluster Annex. Added other minor changes within the document.</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>Included comments from internal AMI group review. New Items: Added Power Factor in Simple Metering cluster. Added Group support in the DR/LC cluster.</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>Included comments from internal AMI group review scattered throughout. A number of field and attribute adds to the clusters.</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>Corrected Document Number issues, otherwise same as Revision 4.</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>Additional changes:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Usual grammar and spelling changes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Load Profile commands have been updated.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Added Attributes to support the latest partial LP interval.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Load Control rules for DR/LC Randomization</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ESP Historical Attrib. changes in Simple Metering cluster</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Changes to the Get Current Price command</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>Grammar, spelling, and formatting changes.</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>Same as Rev 07 except headers and checked in as PDF file.</td>
</tr>
</tbody>
</table>
1 Introduction

1.1 Scope
This profile defines device descriptions and standard practices for applications needed in a residential or light commercial environment. Installation scenarios range from a single home to an entire apartment complex. The key application domains included in this initial version are metering, pricing and demand response and load control applications. Other applications will be added in future versions.

1.2 Purpose
This specification provides standard interfaces and device definitions to allow interoperability among ZigBee devices produced by various manufacturers of electrical equipment, meters and AMI products.
2 References

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

2.1 ZigBee Alliance documents

[R1] ZigBee Document 064321r08, The ZigBee Stack Profile, ZigBee Architectural Sub-Committee of the TSC (TAG)

[R2] ZigBee document 053936r04, ZigBee Cluster Library, Functional Domain: General, ZigBee Application Framework Working Group


[R4] ZigBee document 064309r02, Commissioning Framework

[R5] ZigBee Document 064593, ZigBee Cluster Library

[R6] ZigBee document 03084r00, ZigBee Key Establishment Proposal Certicom

[R7] ZigBee Document 053474r17, The ZigBee Specification, ZigBee Technical Steering Committee (TSC)

[R8] ZigBee Document 074855r03, The ZigBee PRO Stack Profile, ZigBee Architectural Sub-Committee of the TSC (TAG)

[R9] ZigBee Document 074994r08, The AMI Profile Technical Requirements Document, ZigBee AMI Profile Task Group

[R10] ZigBee Document 075297r04, Proposal for Inter-PAN Exchange of Data in ZigBee

2.1.1 External Reference documents


<table>
<thead>
<tr>
<th>Reference</th>
<th>Title</th>
<th>URL</th>
</tr>
</thead>
</table>
3 Definitions

3.1 Conformance levels

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

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

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

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

3.2 ZigBee Definitions

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

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

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

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

Node: Same as a unit.

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

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

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

ZigBee coordinator: An IEEE 802.15.4-2003 PAN coordinator.

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

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

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>AES</td>
<td>Advanced Encryption Standard</td>
</tr>
<tr>
<td>AMI</td>
<td>Advanced Metering Infrastructure or Advanced Metering Initiative</td>
</tr>
<tr>
<td>BPL</td>
<td>Broadband over Powerlines</td>
</tr>
<tr>
<td>CA</td>
<td>Certificate Authority</td>
</tr>
<tr>
<td>CBKE</td>
<td>Certificate-based Key Establishment</td>
</tr>
<tr>
<td>CT</td>
<td>Commissioning Tool</td>
</tr>
<tr>
<td>ECDSA</td>
<td>Elliptic Curve Digital Signature Algorithm</td>
</tr>
<tr>
<td>ECMQV</td>
<td>Elliptic Curve Menezes-Qu-Vanstone</td>
</tr>
<tr>
<td>EMS</td>
<td>Energy Management System</td>
</tr>
<tr>
<td>EPID</td>
<td>Extended PAN Identifier</td>
</tr>
<tr>
<td>ESP</td>
<td>Energy Service Portal</td>
</tr>
<tr>
<td>EUI64</td>
<td>Extended Universal Identifier-64</td>
</tr>
<tr>
<td>GPRS</td>
<td>General Packet Radio Service</td>
</tr>
<tr>
<td>HA</td>
<td>Home Automation</td>
</tr>
<tr>
<td>HAN</td>
<td>Home Area Network</td>
</tr>
<tr>
<td>IHD</td>
<td>In-Home Display</td>
</tr>
<tr>
<td>IVR</td>
<td>Interactive Voice Response</td>
</tr>
<tr>
<td>MAC</td>
<td>Medium Access Control</td>
</tr>
<tr>
<td>MRD</td>
<td>Market Requirements Document</td>
</tr>
<tr>
<td>NAN</td>
<td>Neighborhood Area Network</td>
</tr>
<tr>
<td>PAN</td>
<td>Personal Area Network</td>
</tr>
<tr>
<td>PKKE</td>
<td>Public Key Key Establishment</td>
</tr>
<tr>
<td>PCT</td>
<td>Programmable Communicating Thermostat</td>
</tr>
<tr>
<td>PID</td>
<td>PAN Identifier</td>
</tr>
<tr>
<td>RFD</td>
<td>Reduced Functionality Device</td>
</tr>
<tr>
<td>SAS</td>
<td>Startup Attribute Set</td>
</tr>
<tr>
<td>SKKE</td>
<td>Symmetric Key Key Exchange</td>
</tr>
<tr>
<td>SMAC</td>
<td>Secure Message Authentication Code</td>
</tr>
<tr>
<td></td>
<td>TC</td>
</tr>
<tr>
<td>---</td>
<td>------</td>
</tr>
<tr>
<td>3</td>
<td>TOU</td>
</tr>
<tr>
<td>4</td>
<td>UKE</td>
</tr>
<tr>
<td>5</td>
<td>UTF-8</td>
</tr>
<tr>
<td>6</td>
<td>ZCL</td>
</tr>
<tr>
<td>7</td>
<td>ZDP</td>
</tr>
</tbody>
</table>
5 Profile description

5.1 A ZigBee AMI Network

The ZigBee AMI Technical Requirements Document [R9] details several types of ZigBee networks for metering. These include neighborhood area networks for meters, using ZigBee for sub-metering within a home or apartment, and using ZigBee to communicate to devices within the home. Different installations and utility preferences will result in different network topologies and operation and this profile must allow for these differences. However, each of these networks will operate using the same basic principals to ensure interoperability.

Because of the type of data and control within the AMI network, application security is a key requirement. The application will use link keys which are optional in the ZigBee and ZigBee Pro stack profiles but are required within an AMI network. The installation and use of these keys is described in the security section.

Other devices within a home may also be capable of receiving public pricing information and messages from the metering network. These devices may not have or need all the capabilities required to join an AMI network. Mechanisms are provided to publish public pricing data and messages to these devices without requiring they join the AMI network. These mechanisms are described in the sections on public pricing and messages.

Metering networks are primarily installed by either service personnel but other devices in the network may be added by home owners, or home automation professionals that may not have any ZigBee expertise. Installation concepts must be easy and uniform across multiple OEM vendors.

AMI networks could include both ZigBee 2006 and ZigBee 2007 Pro nodes, it is recommended that the ratio of the nodes should not be 50/50, but the majority of the nodes in the network should be based on one stack profile or the other to get consistent performance. ZigBee AMI certified products must be based upon a ZigBee Compliant Platform (ZCP). If the AMI profile resides in conjunction with a private profile, the product should be ZigBee Manufacturer Specific Profile MSP certified. This additional certification provides a reassurance that the underlying golden stack is behaving properly and the application is not abusive to the network.

Though encouraged, ZigBee AMI products do not require the support of the ZigBee commissioning cluster. They all are required to support “E-mode” of commissioning. E-mode commissioning would classically involve a button push or remote command, but could also use an OEM provided simple tool.

A ZigBee AMI network will not interact with a consumer ZigBee Home Area Network unless a bridge device is used or the HA devices satisfy the AMI security requirements. This is due to the higher security requirements on the AMI network that are not required on a Home network. However, it is expected that Home Automation devices that are extended to include the AMI requirements can still operate in a home network.

A ZigBee AMI makes possible networks such as the following:
Utility Private HAN might include an in-home display, or a load control device working in conjunction with energy service portal, but it would not include any customer controlled devices.

Utility Private ZigBee network might also be used as a NAN, where ZigBee provided the primary communications for an AMI deployment.
Figure 3 Customer Private HAN

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

5.2 ZigBee stack profile

Products that conform to this specification shall use stack profile number 0x01 or profile 0x02, as defined in [R1]. In addition to the requirements specified in [R1], the following requirements are mandatory for this application profile.

- Support for Application link keys is required.
- Fragmentation is required.
- In their normal operating state, ZigBee end devices shall poll no more frequently than once every 7.5 seconds except where this specification indicates otherwise for a particular device description, or under the following conditions. ZigBee end devices may operate with a higher polling rate during commissioning, network maintenance, alarm states, and for short periods after transmitting a message to allow for acknowledgements and or responses to be received quickly, but they must return to the standard rate indicated previously during normal operation. It is recommended that ZigBee end devices poll much less frequently than once per 7.5 seconds, especially when the device normally only communicates due to user interaction.

5.2.1 ZigBee Routing Table Size Recommendations

- In a neighborhood area network, data flow to the AMI server should use Many to One Routing when on a ZigBee Pro stack to minimize route table size. In a ZigBee network this data flow is either handled by the tree or routing table sizes must be increased to account for routes from all devices in the network to the AMI server. Please be aware the Many to One relationship will impact the size of the AMI Server source routing tables and is implementation specific.
- If an AMI device is communicating into a ZigBee AMI network, the routing table size should account for all expected AMI devices within a premise.
5.2.2 ZigBee Coordinator and Trust Center Recommendations

- In an AMI based HAN network the ESP should act as the coordinator and trust center of the network.
- In an AMI based NAN the backhaul point is likely to be the coordinator and trust center.

5.3 Startup Attribute Set (SAS)

- In order to insure interoperability, all ZigBee AMI devices should implement compatible Startup Attribute Sets (SAS). This does not mean that set must be modifiable through a commissioning cluster, but that the device must internally implement these stack settings to insure compatibility and consistent user experience. The start up set parameters described by the commissioning cluster in [R4] provide a good basis to specify an AMI start up set.

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

5.3.1 Start Up Parameters

- Short Address – 0xFFFF
- E PANID – 0x0000000000000000 or installer specified
- PAN ID – 0xFFFF or installer specified
- 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 join command is indicated.
  - 0 (Zero) if commissioned. Indicates that the device should consider itself a part of the network indicated by the ExtendedPANId attribute. In this case it will not perform any explicit join or rejoin operation.
- Trust Center Address – 0x0000000000000000 or installer specified. Please note: Identifying or establishing the Trust Center as a device other than the Coordinator is an optional stack profile feature. Since this is an implementation specific issue, installation tools and setting address is managed by the OEM vendors.
- Master Key – not used, high security is not used in this profile.
- Link Key – 0x0000000000000001 if the Key Establishment Cluster is being used to install a link key
  - Installer provided if using preconfigured link keys
- Network Key – 0x0000000000000001 if no pre-installed key present.
- Use Insecure Join – 0x02 (False) - flag that enables the use of insecure join as a fallback case at startup time. (This only applies to 2007 stack profile devices).
5.3.2 Join Parameters

ScanAttempts – At boot time or when instructed to join a network, the device should make 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 (2006) devices can only scan for their original PAN ID).

TimeBetweenScans – (1 second) determines the number of seconds between each
unsuccessful 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 itself disconnected, 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.

5.3.3 Security Parameters

SecurityTimeoutPeriod – set by stack profile.

TrustCenterNetworkKey – the Trust center will pick the network key. ZigBee AMI devices
shall depend on either pre-configured keys to be commissioned or the use of the Key Establishment
Cluster to install a key. ZigBee AMI networks will not generally send keys in the clear.

5.3.4 End Device Parameters

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

5.3.5 Link Status Parameters

LinkStatusPeriod - Set by stack profile.
RouterAgeLimit - Set by stack profile.
RepairThreshold - Set by stack profile.

5.3.6 Concentrator Parameters

ConcentratorFlag - configures device to be a concentrator. This would typically be a part of
an OEM NAN and is not required to be on an AMI certified device or configurable by 3rd
party tool. If an OEM does make a device that can be a concentrator, it does not have to be
configurable in any standardized way.

ConcentratorRadius – 12 (twelve) – OEMs that make a concentrator product will set the max
concentrator radius to this value.
ConcentratorDiscoveryTime – Set by stack profile. This is how soon nodes should reply to a concentrator after hearing a route request command.

### 5.3.7 APS Transport Parameters

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.

### 5.3.8 Binding Parameters

EndDeviceBindTimeout – 60 seconds - timeout value for end device binding. End Device binding is set by coordinator.

### 5.4 AMI Security

The AMI network requires the use of the Key Establishment Cluster (Annex C) for installation and updating of link keys.

There are two possible security settings for joining within an AMI network, one using preinstalled link keys and one using a temporary link key prior to using the Key Establishment Cluster. In either of these cases it is not desirable to send keys in the clear and it is generally prohibited unless they are then updated throughout the network securely.

To ensure clear compatibility, each of these methods is detailed here.

#### 5.4.1 Joining with Preinstalled Link Keys

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

1. Link keys are installed in each device to join the utility network and these link keys are known to the existing network through some vendor specific means.

2. The link key for a device that is to be joined is provided to the local trust center through an out of band means.

3. Permit joining is turned on in the network.

4. The device joins the network and is sent the network key encrypted with the preinstalled link key.

5. The trust center must update the link key with the joining device using the Key Establishment Cluster after completion of joining.

6. The trust center of the network has the option of updating link keys with devices as desired by the application using the Key Establishment Cluster.

7. If devices leave the network, the trust center shall update the network key.
5.4.2 Joining using the Key Establishment Cluster

When using the Key Establishment Cluster to establish a higher level of authentication and authorization the following steps are used:

1. Devices have a random number preinstalled that is known to the existing network through some vendor specific means. This random number will be used as part of a temporary link key to join the network.

2. Permit joining is turned on in the network.

3. The device joins the network and is sent the network key encrypted with a temporary link key that is a cryptographic hash of the random number and devices EUI 64.

4. Using the temporary link key, the device and the trust center use the Key Establishment Cluster to establish authorization and authentication.
   a. If the trust center does not complete the authorization and authentication within 15 minutes, the device is sent a leave command and the trust center removes it from its tables.

5. If this Key Establishment Cluster public key process is successful a permanent link key is established in the device.

6. The trust center has the option of updating link keys using the Key Establishment Cluster as desired by the application.

7. The trust center has the option of updating network keys using the existing link keys as desired by the application.

5.5 Public Pricing

It is required that some ZigBee AMI devices respond to requests for public pricing information from devices that are not on the ZigBee AMI network and do not share the same security settings as the ZigBee AMI devices. Only those devices expected to support and communicate this public pricing information must implement this functionality. Devices that support public pricing must support the price cluster within the ZigBee AMI profile. The data from this ZigBee AMI profile is used as the basis for the public pricing broadcast. In other words, the ZigBee AMI devices receive pricing information securely over the ZigBee network but broadcast it anonymously and publicly using the anonymous Inter-PAN transmission mechanism outlined in Annex B. This public pricing is published as a 1 hop broadcast on the existing channel of the ZigBee AMI network. A device that receives a message formatted as pricing request from the Price Cluster will respond with a public and anonymous pricing message. This message is sent as a 1 hop broadcast formatted as a price response command from the Price Cluster.

5.6 Other AMI Requirements and Best Practices

- Preferred Channels - 11, 14, 15, 19, 20, 24, 25
  When forming a new network, or scanning to join a network, AMI devices should do channel scans using the above channel mask before scanning the rest of the channels in order to avoid the most commonly used WiFi channels. This is to improve the user experience during installation (quicker joining) and possibly improve bandwidth (on average).

- Broadcast Policy
Except for public pricing, broadcasts are to be strongly discouraged for AMI devices. Devices are limited to a maximum broadcast frequency of one broadcast per second and strongly encouraged to exercise broadcasts much less frequently.

**Frequency Agility**

Frequency Agility would only be officially exercised in a network by an OEM system controller, or higher functioning device (ESP, aggregator, installation tool, etc...). Devices may support frequency agility hooks to be commanded to “go to channel X”. Devices that do not support frequency agility must implement the orphan join feature so that they will find a network that has changed channels.

**Key Updates**

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

### 5.7 Coexistence and Interoperability With HA Devices

It is desirable to allow interoperability of HA and AMI devices where practical. However, it is undesirable to publicly share keys during the joining process or share private information over a less secure network. HA devices that only provide functionality for receiving network keys in the clear during a join process cannot be used in an AMI network. These devices can receive public pricing information as described above. HA devices may also be extended with AMI clusters and be certified as HA and AMI capable. These devices can operate either in an HA network or an AMI network.

### 5.8 Device descriptions

Device descriptions specified in this profile are summarized in Table 2 along with their respective Device IDs. The devices are organized according to the end application areas they address. A product that conforms to this specification shall implement at least one of these device descriptions and shall also include the device descriptions corresponding to all applications implemented on the product where a standard device description is specified in this profile. For example, if a product implements both a thermostat and an In-Premise Display, then the thermostat and In-Premise Display device descriptions must both be supported.

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

**Table 2 – Devices specified in the AMI profile**

<table>
<thead>
<tr>
<th>Device Description</th>
<th>Device ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range Extender</td>
<td>0x0008</td>
</tr>
<tr>
<td>Energy Service Portal</td>
<td>0x0500</td>
</tr>
<tr>
<td>Meter</td>
<td>0x0501</td>
</tr>
</tbody>
</table>
5.9

5.10 ZigBee Cluster Library (ZCL)

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.

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. The attributes that a particular cluster is
capable of reporting are listed in the ZCL specification for each cluster. Products shall support the
reporting mechanism for all attributes specified in the ZCL that the product implements within a given
cluster. The minimum reporting interval specified in the ZCL [R3] shall be set to a value greater than
or equal to 0x0001. The maximum reporting interval should be set to 0x0000 by default, and if it is set
to a non-zero value it shall be set to a value greater than or equal to 0x003C and greater than the value
of the minimum reporting interval. These settings will restrict the attributes from being reported more
often than once every second if the attribute is changing quickly and at least once every minute if the
attribute does not change for a long time. It is recommended that the minimum reporting interval be set
to a higher value whenever the application can tolerate it. It is recommended that the maximum
reporting interval be set to a much greater value to avoid unnecessary traffic.

5.11 Cluster List

The clusters used in this profile are listed in Table 3. The clusters are listed according the functional
domain they belong to in the ZCL and indicate the additional new AMI clusters. The existing
corresponding ZCL General cluster identifiers can be found in the ZCL Foundation specification [R3].

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

Most clusters include optional attributes. The application designer must be aware that optional
attributes may not be implemented on a particular device. It is the responsibility of a device’s
application to discover and deal with unsupported attributes on other devices.

It is expected that clusters will continue to be developed in the ZCL that will be useful in this profile.
In many cases, new clusters will be organized into new device descriptions that are separate from those
currently defined. There may also be situations where it makes sense to add clusters as optional or
possibly even mandatory elements of existing device descriptions.
Manufacturer-specific clusters may be added to any device description in this profile as long as they follow the specifications given in the ZCL Foundation specification [R3].

Table 3 – Clusters used in the AMI profile

<table>
<thead>
<tr>
<th>Functional Domain</th>
<th>Cluster Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
<td>Basic</td>
</tr>
<tr>
<td>General</td>
<td>Groups</td>
</tr>
<tr>
<td>General</td>
<td>Alarms</td>
</tr>
<tr>
<td>General</td>
<td>Time</td>
</tr>
<tr>
<td>General</td>
<td>Key Establishment</td>
</tr>
<tr>
<td>AMI</td>
<td>Price</td>
</tr>
<tr>
<td>AMI</td>
<td>Demand Response and Load Control</td>
</tr>
<tr>
<td>AMI</td>
<td>Simple Metering</td>
</tr>
<tr>
<td>AMI</td>
<td>Message</td>
</tr>
<tr>
<td>AMI</td>
<td>Registration</td>
</tr>
<tr>
<td>AMI</td>
<td>AMI Tunneling (Complex Metering)</td>
</tr>
<tr>
<td>AMI</td>
<td>Pre-Payment</td>
</tr>
</tbody>
</table>

5.11.1 General Clusters

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

5.11.1.1 ZCL Time Cluster and Time Synchronization

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

It’s highly recommended that the processes for synchronizing time be as network friendly as possible to eliminate the possibility of excessive traffic. To this end, time synchronization should be limited to when clock drift or changes with individual products necessitate an update, or when a critical event or a process that is highly time dependant is about to take place.

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

5.12.1 Demand Response and Load Control Cluster

5.12.1.1 Overview

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

5.12.1.2 Server

5.12.1.2.1 Dependencies

Events carried using this cluster include a timestamp with the assumption that target devices maintain a real time clock. Devices can acquire and synchronize their internal clocks via the ZCL Time server. If a device does not support a real time clock it’s assumed the device will interpret and utilize the “Start Now” value within the Time field.

5.12.1.2.2 Attributes

There are no attributes for the Demand Response and Load Control Cluster server.

5.12.1.2.3 Commands Generated

The command IDs generated by the Demand Response and Load Control cluster server are listed in Table 4.

<table>
<thead>
<tr>
<th>Command identifier field value</th>
<th>Description</th>
<th>Mandatory / Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>Load Control Event</td>
<td>M</td>
</tr>
<tr>
<td>0x01</td>
<td>Cancel Load Control Event</td>
<td>M</td>
</tr>
<tr>
<td>0x02</td>
<td>Cancel All Load Control Events</td>
<td>M</td>
</tr>
<tr>
<td>0x03 – 0xff</td>
<td>Reserved</td>
<td></td>
</tr>
</tbody>
</table>

5.12.1.2.3.1 Load Control Event(

5.12.1.2.3.1.1 Payload Format

The Load Control Event command payload shall be formatted as illustrated in Figure 4
### Field Name

<table>
<thead>
<tr>
<th>Bits</th>
<th>32</th>
<th>24</th>
<th>32</th>
<th>16</th>
<th>8</th>
<th>8</th>
</tr>
</thead>
</table>

#### Data Type

<table>
<thead>
<tr>
<th>Field</th>
<th>Data Type</th>
<th>Data Type</th>
<th>Data Type</th>
<th>Data Type</th>
<th>Data Type</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Issuer Event ID</td>
<td>Unsigned 32-bit integer</td>
<td>Unsigned 24-bit integer</td>
<td>UTCTime</td>
<td>Unsigned 16-bit integer</td>
<td>Unsigned 8-bit integer</td>
<td>Cooling Temperature Offset</td>
</tr>
<tr>
<td>Device / Group Class</td>
<td>start Time</td>
<td>Duration In Minutes</td>
<td>Criticality Level</td>
<td>Heating Temperature Offset</td>
<td>Temperature Offset</td>
<td></td>
</tr>
</tbody>
</table>

### Table 4 – Format of the Load Control Event payload

<table>
<thead>
<tr>
<th>16</th>
<th>16</th>
<th>8</th>
<th>8</th>
<th>8</th>
</tr>
</thead>
</table>

- **Cooling Temperature Set Point**: Indicates the current set point for cooling.
- **Heating Temperature Set Point**: Indicates the current set point for heating.
- **Average Load Adjustment Percentage**: This field indicates the percentage of load adjustment needed.
- **Duty Cycle**: Indicates the percentage of time the load is active.
- **Event Control**: Indicates the control action to be taken.

### 5.12.1.2.3.1.2 Payload Details

- **Issuer Event ID**: 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.

### Device Class

- **Bit 0 - HVAC compressor or furnace**
- **Bit 1 - Strip Heat**
- **Bit 2 - Water Heater**
- **Bit 3 - Pool Pump/Spa/Jacuzzi**
- **Bit 4 - Smart Appliances**
- **Bit 5 - Irrigation Pump**
- **Bit 6 - Managed (C&I) loads**
- **Bit 7 - Simple misc. (Residential On/Off) loads**

---

Note: All fields must be present in the payload. Optional fields will be marked with specific values to indicate they are not being used.
Bit 8 - Exterior Lighting
Bit 9 - Interior Lighting
Bit 10 - Electric Vehicle
Bit 11 - Generation Systems
Bits 12 to 19 - Future use
Bits 20 to 23 – Utility defined Group identifier. Please refer to section 5.12.1.3.2.1 Client Cluster Attributes for further definition and usage of these bits.

Start Time: UTC Timestamp representing when the event is scheduled to start. A start time of 0xffffffff is a special time denoting “now.”

Duration In Minutes: Duration of this event in number of minutes.

Criticality Level: This field defines the level of criticality of this event. The action taken by end devices for this event can strictly be based on this value or can be combined with one or more of the provided hints such as Average Load Adjustment Percentage, Duty Cycle, Cooling Temperature Offset, Heating Temperature Offset, Cooling Temperature Set Point or Heating Temperature Set Point.

<table>
<thead>
<tr>
<th>Criticality Level</th>
<th>Level Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Reserved</td>
</tr>
<tr>
<td>1</td>
<td>Green</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>7</td>
<td>Emergency</td>
</tr>
<tr>
<td>8</td>
<td>Planned Outage</td>
</tr>
<tr>
<td>9</td>
<td>Service Disconnect</td>
</tr>
<tr>
<td>0x0A to 0x0F</td>
<td>Utility Defined</td>
</tr>
<tr>
<td>0x10 to 0xFF</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

The criticality level 0x0 and 0x10 to 0xFF are reserved for future profile changes and not used.

Green” event, level 0x01, may be used to denote that the energy delivered uses an abnormal amount from non-”green” sources. Participation in this event is voluntary.

The criticality levels 0x02 through 0x06 (Level 1 through 5) indicate progressively increasing levels of load reduction are being requested by the utility. Participation in these events is voluntary.

The criticality level 0x07 is used to indicate an “Emergency” event. Participation in this event is mandatory, as defined by the utility. The expected response to this event is termination of all non-essential energy use, as defined by the utility. Exceptions to participation in this event type must be managed by the utility.
The criticality level 0x08 is used to indicate a “Planned Outage” event. Participation in this event is mandatory, as defined by the utility. The expected response to this event is termination of delivery of all non-essential energy, as defined by the utility. Exceptions to participation in this event type must be managed by the utility.

The criticality level 0x09 is used to indicate a “Service Disconnect” event. Participation in this event is mandatory, as defined by the utility. The expected response to this event is termination of delivery of all non-essential energy, as defined by the utility. Exceptions to participation in this event type must be managed by the utility.

Levels 0x0A to 0x0F are available for Utility Defined criticality levels.

**Cooling Temperature Offset:** Requested offset to apply to the normal cooling setpoint at the time of the start of the event in + 0.1 degrees C.

**Heating Temperature Offset:** Requested offset to apply to the normal heating setpoint at the time of the start of the event in + 0.1 degrees C.

The Cooling and Heating Temperature Offsets represent a temperature change (Delta Temperature) that will be applied to both the associated heating and cooling set points. The temperature offsets (Delta Temperatures) will be calculated per the Local Temperature attribute in the Thermostat Cluster. The calculated temperature will be interpreted as the number of degrees to be added to the cooling set point and subtracted from the heating set point.

Each offset represents the temperature offset (Delta Temperature) in degrees Celsius, as follows: Delta Temperature = 10 x temperature in degrees Celsius. Where 0.00°C <= temperature <= 40.00 °C, corresponding to a Temperature in the range 0x00 to 0x0FE. The maximum resolution this format allowed is 0.1 °C.

A DeltaTemperature of 0xFF indicates that the temperature offset is not used.

If a temperature offset is sent that causes the heating or cooling temperature set point to exceed the limit boundaries that are programmed into the thermostat, the thermostat will respond by setting the temperature at the limit.

**Cooling Temperature Set Point:** Requested cooling set point in 0.01 degrees Celsius.

**Heating Temperature Set Point:** Requested heating set point in 0.01 degrees Celsius.

Cooling and heating temperature set points will be defined and calculated per the LocalTemperature attribute in the Thermostat Cluster [ZCL reference].

These fields represent the temperature in degrees Celsius, as follows:

Cooling and Heating Temperature Set Point = 100 x temperature in degrees Celsius.

Where -273.15°C <= temperature <= 327.67°C, corresponding to a Cooling and/or Heating Temperature Set Point in the range 0x954D to 0x7FFF.

The maximum resolution this format allows is 0.01°C.

A Cooling or Heating Temperature Set Point of 0xFFFF indicates that the temperature set point is not used.

If a temperature is sent that exceeds the temperature limit boundaries that are programmed into the thermostat, the thermostat will respond by setting the temperature at the limit.

When both a Temperature Offset and a Temperature Set Point are provided, the thermostat may use either as defined by the equipment manufacturer.
Average Load Adjustment Percentage: Defines a maximum energy usage limit as a percentage of an OEM specific average energy usage. The load adjustment percentage is added to 100% creating a percentage limit applied to the OEM specific average energy usage. A -10% load adjustment percentage will establish an energy usage limit equal to 90% of the OEM specific average energy usage. Each load adjustment percentage is referenced to the OEM specific average energy usage. There are not cumulative effects.

This field applies only to the time period defined by start time and duration.

The range of this field is -100 to 0 with a resolution of 1 percent. A -100% value equals a total load shed. A 0% value will limit the energy usage to the OEM specific average energy usage.

A value of 0xFF indicates the field is not used. All other values are reserved for future use.

Duty Cycle: Defines the maximum On state duty cycle as a percentage of time. Example, if the value was 80, the device would be in an “on state” for 80% of the time for the duration of the event. Range of the value is 0 to 100. A value of 0xFF indicates the field is not used. All other values are reserved for future use.

Event Control: Identifies additional control options for the event.

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0= Randomize start time, 1=Randomized Start not Applied</td>
</tr>
<tr>
<td>1</td>
<td>0= Randomize End time, 1=Randomized End not Applied</td>
</tr>
</tbody>
</table>

Note: The randomization attribute will be used in combination with two bits to determine if the Event Start and Stop Time is randomized. By default devices will randomize the start and stop of an event.

Bits 2 to 6: Reserved.

5.12.1.2.3.1.3 When generated

This command is generated when the ESP wants to control one or more load control device(s), usually as the result of an energy curtailment command from the AMI network.

5.12.1.2.4 Commands Received

The server receives the cluster specific commands detailed in 5.12.1.3.1.

5.12.1.2.4.1 Cancel Load Control Event()
### Figure 5 – Format of the Cancel Load Control Event payload

Note: All fields must be present in the payload. Optional fields will be marked with specific values to indicate they are not being used.

#### 5.12.1.2.4.1.2 Payload Details

**Issuer Event ID**: 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.

**Device Class**: Bit encoded field representing the Device Class and grouping to apply the current Load Control Event. Each bit, if set singly or in combination, indicate(s) the group/class device(s) needing to take action. The encoding of this field is:

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>HVAC compressor or furnace</td>
</tr>
<tr>
<td>1</td>
<td>Strip Heat</td>
</tr>
<tr>
<td>2</td>
<td>Water Heater</td>
</tr>
<tr>
<td>3</td>
<td>Pool Pump/Spa/Jacuzzi</td>
</tr>
<tr>
<td>4</td>
<td>Smart Appliances</td>
</tr>
<tr>
<td>5</td>
<td>Irrigation Pump</td>
</tr>
<tr>
<td>6</td>
<td>Managed (C&amp;I) loads</td>
</tr>
<tr>
<td>7</td>
<td>Simple misc. (Residential On/Off) loads</td>
</tr>
<tr>
<td>8</td>
<td>Exterior Lighting</td>
</tr>
<tr>
<td>9</td>
<td>Interior Lighting</td>
</tr>
<tr>
<td>10</td>
<td>Electric Vehicle</td>
</tr>
<tr>
<td>11</td>
<td>Generation Systems</td>
</tr>
<tr>
<td>12</td>
<td>Bits 12 to 19 - Future use</td>
</tr>
</tbody>
</table>
| 20  | Bits 20 to 23 – Utility defined Group identifier. Please refer to section 5.12.1.3.2.1 Client Cluster Attributes for further definition and usage of these bits.

**Cancel Control**:

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
</table>

To be used when the Event is currently in process and a cancel command is received. A value of Zero (0) indicates that the event should end using the rules established by the device and the event when transmitted to the device (Example: Randomized ending will be observed.) A value of one (1) indicates that the event should be terminated immediately.

1 to 7 Reserved

Effective Time: UTC Timestamp representing when the canceling of the event is scheduled to start. A start time of 0xffffffff is a special time denoting “now.”

5.12.1.2.5 Commands Received
The server receives the cluster specific commands detailed in 5.12.1.3.1.

Note: If the Cancel command is received after the event has occurred, the device shall reply using the “Report Event Status Cluster” with an Event Status of “Load Control Event command Rejected”.

5.12.1.2.5.1 Cancel All Load Control Events( )
5.12.1.2.5.1.1 Payload Format
The Cancel All Load Control Events command payload shall be formatted as illustrated in Figure 6.

<table>
<thead>
<tr>
<th>Bits</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field Name</td>
<td>Cancel Control</td>
</tr>
<tr>
<td>Data Type</td>
<td>Unsigned 8-bit integer</td>
</tr>
</tbody>
</table>

Figure 6 – Format of the Cancel All Load Control Events payload

5.12.1.2.5.1.2 Payload Details
Cancel Control:

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>To be used when the Event is currently in process and a cancel command is received. A value of Zero (0) indicates that the event should end using the rules established by the device and the event when transmitted to the device (Example: Randomized ending will be observed.) A value of one (1) indicates that the event should be terminated immediately.</td>
</tr>
<tr>
<td>1 to 7</td>
<td>Reserved</td>
</tr>
</tbody>
</table>
5.12.1.2.5.1.3 When generated

This command is generated when the ESP wants to cancel all events for control device(s).

5.12.1.2.6 Commands Received

The server receives the cluster specific commands detailed in 5.12.1.3.3.1. The Cancel All Load Control Events command is processed by the device as if individual Cancel Load Control Event commands were received for all of the currently stored events in the device. The device will respond with a “Report Event Status Cluster” for each individual load control event canceled.

5.12.1.3 Client

5.12.1.3.1 Dependencies

None.

5.12.1.3.2 Client Cluster Attributes

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Name</th>
<th>Type</th>
<th>Range</th>
<th>Access</th>
<th>Default</th>
<th>Man. / Opt.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>UtilityDefinedGroup</td>
<td>Unsigned 8 bit Integer</td>
<td>0x00 to 0x0F</td>
<td>Read/Write</td>
<td>-</td>
<td>M</td>
</tr>
<tr>
<td>0x01</td>
<td>StartRandomizeMin</td>
<td>Unsigned 6 bit Integer</td>
<td>0x00 to 0x3C</td>
<td>Read/Write</td>
<td>0x1E</td>
<td>M</td>
</tr>
<tr>
<td>0x02</td>
<td>StopRandomizeMin</td>
<td>Unsigned 6 bit Integer</td>
<td>0x00 to 0x3C</td>
<td>Read/Write</td>
<td>0x1E</td>
<td>M</td>
</tr>
<tr>
<td>0x3 to 0xF</td>
<td>Reserved</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5.12.1.3.2.1 Utility Defined Group Attribute

The UtilityDefinedGroup represents provides a method for utilities to assign devices to groups. The valid range for this attribute is 0x00 to 0x0F where 0x00 indicates no group specified and remaining groups enumerated from 1 to 15 (0x0F). The definition of the groups and their assigned values is outside the scope of this specification.

5.12.1.3.2.2 Start Randomization Minutes

The StartRandomizedMin represents the number of minutes to be used when randomizing the start of an event. The valid range for this attribute is 0x00 to 0x3C where 0x00 indicates start event randomization is not performed.

5.12.1.3.2.3 End Randomization Minutes

The EndRandomizedMin represents the number of minutes to be used when randomizing the End of an event. The valid range for this attribute is 0x00 to 0x3C where 0x00 indicates end event randomization is not performed.
5.12.1.3.3 Commands Generated

The command IDs generated by the Demand Response and Load Control cluster client are listed in Table 7.

<table>
<thead>
<tr>
<th>Command identifier field value</th>
<th>Description</th>
<th>Mandatory / Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>Report Event Status</td>
<td>M</td>
</tr>
<tr>
<td>0x01 – 0xff</td>
<td>Reserved</td>
<td></td>
</tr>
</tbody>
</table>

5.12.1.3.3.1 Report Event Status(

5.12.1.3.3.1.1 Payload Format

The Report Event Status command payload shall be formatted as illustrated in Figure 7.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Data Type</th>
<th>Bits</th>
<th>32</th>
<th>32</th>
<th>32</th>
<th>8</th>
<th>16</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Issuer Event ID</td>
<td>Unsigned 16-bit integer</td>
<td>8</td>
<td>16</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Event Status</td>
<td>8-bit enumeration</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Start Time</td>
<td>UTCTime</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Criticality Level Applied</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cooling Temperature Set Point Applied</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Heating Temperature Set Point Applied</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Continuation of payload...

<table>
<thead>
<tr>
<th>8</th>
<th>8</th>
<th>8</th>
<th>128</th>
</tr>
</thead>
<tbody>
<tr>
<td>Applied Average Load Adjustment Percentage</td>
<td>Applied Duty Cycle</td>
<td>Event Control</td>
<td>Signature</td>
</tr>
<tr>
<td>Signed 8-bit integer</td>
<td>Unsigned 8-bit integer</td>
<td>Unsigned 8-bit integer</td>
<td>Octet string</td>
</tr>
</tbody>
</table>

Figure 7 – Format of the Report Event Status Command payload

5.12.1.3.3.1.2 Payload Details

Issuer Event ID: Unique identifier generated by the Energy provider. The value of this field allows matching of Event reports with a specific Demand Response and Load Control event. It’s expected the value contained in this field is a unique number managed by upstream systems or a UTC based time stamp (UTCTime data type) identifying when the Load Control Event was issued.
Event Status:
1 = Load Control Event command received
2 = Event started
3 = Event completed
4 = User has chosen to OptOut
5 = User has chosen to OptIn
6 = The event has been cancelled
7 = The event has been superseded
0xFE = Load Control Event command Rejected.

Start Time: UTC Timestamp representing when the event was established in the device.

Criticality Level Applied: Criticality Level value applied by the device, see the corresponding field in the Load Control Event Command for more information.

Cooling Temperature Set Point Applied: Cooling Temperature Set Point value applied by the device, see the corresponding field in the Load Control Event Command for more information. The value 0x8000 means that this field has not been used by the end device.

Heating Temperature Set Point Applied: Heating Temperature Set Point value applied by the device, see the corresponding field in the Load Control Event Command for more information. The value 0x8000 means that this field has not been used by the end device.

Applied Average Load Adjustment Percentage: Average Load Adjustment Percentage value applied by the device, see the corresponding field in the Load Control Event Command for more information.

Applied Duty Cycle: Defines the maximum On state duty cycle applied by the device.

Event Control: Identifies additional control options for the event.

Signature: Signature used to implement non-repudiation.

5.12.1.3.3.1.3 When generated
This command is generated when the client device detects a change of state for an active Load Control event.

5.12.1.3.4 Commands Received
The client receives the cluster specific commands detailed in 5.12.4.2.3.1.

5.12.1.3.5 Attribute reporting
Not used.

5.12.1.4 Application Guidelines
The criticality level is sent by the utility to the load shed control to indicate how much load reduction is requested. The utility is not required to use all of the criticality levels that are described in this specification. A load shed control is not required to provide a unique response to each criticality level that it may receive.

The Average Load Adjustment Percentage, temperature offsets, and temperature set points are used by load shed devices and energy management systems on a "voluntary" or "optional" basis. These devices are not required to use the values that are provided by the utility. They are provided as a
recommendation by the utility. Any exceptions that dictate the use of these values as “required” or "mandatory" will be defined in this specification.

The Average Load Adjustment Percentage is sent by the utility to the load shed control to indicate how much load reduction is requested. Response of a load shed control to this information is not mandatory.

The Duty Cycle is sent by the utility to the load shed control to indicate the maximum “On state” for a device. Response of a load shed control to this information is not mandatory.

The cooling temperature offset is sent by the utility to the load shed control to indicate how much indoor cooling temperature offset is requested. Response of a load shed control to this information is not mandatory.

The heating temperature offset is sent by the utility to the load shed control to indicate how much indoor heating temperature offset is requested. Response of a load shed control to this information is not mandatory.

The cooling temperature may be sent by the utility to the load shed control to indicate the indoor cooling temperature setting that is requested. Response of a load shed control to this information is not mandatory.

The heating temperature may be sent by the utility to the load shed control to indicate the indoor heating temperature setting that is requested. Response of a load shed control to this information is not mandatory.

Note: The most recent Load Control Event supersedes any previous Load Control Event command for the set of Device Classes and groups for a given time.

5.12.1.4.1 Load Control Rules, Server

5.12.1.4.1.1 Load Control Server, Identifying Use of SetPoint and Offset Fields

All fields in the payload must be populated although the optional usage of the Heating and Cooling Temperature Set Points and Heating and Cooling Temperature Offsets can be signaled by using the value 0x8000. When any of those four fields are filled with a value of 0x8000, they are to be ignored.

5.12.1.4.1.2 Load Control Server, Editing of Scheduled Events

Editing of a scheduled demand response event is not allowed. Editing of an active demand response event is not allowed. Nested events and overlapping events are not allowed. The current active event will be terminated if a new event is started.

5.12.1.4.2 Load Control Rules, Client

5.12.1.4.2.1 Start and Stop Randomization

When shedding loads (turning a load control device off), the load control device will optionally apply start time randomization based on the values specified in the Event Control Bits and the Clients Start Randomization Minutes attribute. By default devices will apply a random delay between zero and 30 minutes. Devices will support randomization delays of up to 60 minutes.
When ending a load control event, the load control device will support the same randomization features
as provided in the start load control event.

5.12.1.4.3 Editing of DR Control Parameters

In load shed controls and energy management systems, editing of the demand response control
parameters while participating in an active demand response event is not allowed.

5.12.1.4.4 Response to Price Events + Load Control Events

The residential system’s response to demand response events will be added to the residential system’s
response to price driven events. Demand response events which require that the residential system is
turned off have priority over price driven events. Demand response events which require that the
residential system go to a fixed setting point have priority over price driven events.

5.12.1.4.5 Thermostat/HVAC Controls

A residential HVAC system will be allowed to change mode, from off to Heat, off to Cool, Cool to
Heat, or Heat to Cool, during a voluntary event which is currently active. The HVAC control must
acknowledge the event, as if it was operating, in that mode, at the start of the event. The HVAC
control must obey the event rules that would have been enforced if the system had been operating in
that mode at the start of the active event.

An event override message, “opt-out”, will be sent by the load shed control or energy management
system if the operator chooses not to participate in a demand response event by taking action to
override the programmed demand reduction response. The override message will be sent at the start of
the event. In the case where the event has been acknowledged and started, the override message will
be sent when the override occurs.

5.12.1.4.6 Demand Response & Load Control Examples

The following example in Figure 8 depicts the transactions that would take place for two events, one
that is successful and another that is overridden by the user.
Head end  ESP  AMI Device

Load Curtailment ID=6

LoadControlEvent ( EventID=6, StartTime=1:00pm, Duration=60 min ...)

ReportEventStatus ( EventID=6, EventCode=CommandReceived ...)

ReportEventStatus ( EventID=6, EventCode=EventStarted )

ReportEventStatus ( EventID=6, EventCode=EventCompleted )

Load Curtailment ID=7

LoadControlEvent ( EventID=7, StartTime=3:00pm, Duration=60 min ...)

ReportEventStatus ( EventID=7, EventCode=CommandReceived ...)

ReportEventStatus ( EventID=7, EventCode=EventStarted )

ReportEventStatus ( EventID=7, EventCode=UserOverride )

ReportEventStatus ( EventID=7, EventCode=EventCompleted )
Figure 8 - Example of both a successful and an overridden Load Curtailment event

The following in Figure 9 depicts the transactions that would take place when an event is superseded by an event that is eventually cancelled.
Load Curtailment ID=8

Load Curtailment ID=9

Load Curtailment Cancel ID=9

Head end  ESP  AMI Device

LoadControlEvent (EventID=8, StartTime=1:00pm, Duration=60 min ...)

ReportEventStatus (EventID=8, EventCode=CommandReceived ...)

ReportEventStatus (EventID=8, EventCode=Superseded)

ReportEventStatus (EventID=9, EventCode=EventStarted)

LoadCancelEvent (EventID=9)

ReportEventStatus (EventID=9, EventCode=Cancelled)
5.12.2 Simple Metering Cluster

5.12.2.1 Overview

The Simple Metering Cluster provides a mechanism to retrieve usage information from Electric, Gas, Water, and potentially Thermal metering devices. These devices can operate on either battery or mains power, and can have a wide variety of sophistication. The Simple Metering Cluster is designed to provide flexibility while limiting capabilities to a set number of metered information types. More advanced forms or data sets from metering devices will be supported in the Complex Metering Cluster.

The following figures identify three configurations as examples utilizing the Simple Metering Cluster.
In the above example, the metering device is the source of information provided via the Simple Metering Cluster Server.
In the above example, the metering device is running on battery power and its duty cycle for providing information is unknown. It’s expected the ESP will act like a mailbox for the metering device data, allowing other AMI devices to gain access to the metering device’s data (provided via an image of its Simple Metering Cluster).

![Diagram](image)

**Figure 12 ESP Model w/Integrated Metering Device**

The above example is much like the previous example in Figure 4, where the external metering device is running on battery power and its duty cycle for providing information is unknown. It’s expected the ESP will act like a mailbox for the external metering device data, allowing other AMI devices to gain access to the metering device’s data (provided via an image of its Simple Metering Cluster). Since the ESP can also contain an integrated metering device where its information is also conveyed through the Simple Metering Cluster, each device (external metering device mailbox and integrated meter) will be available via independent EndPoint IDs. The other AMI devices must understand the ESP configurations.

### 5.12.2.2 Server

#### 5.12.2.2.1 Dependencies

The Simple Metering Cluster will require the use of the ZCL to access attributes and provide subscribed reporting of metered data.

#### 5.12.2.2 Attributes

For convenience, the attributes defined in this specification are arranged into sets of related attributes; each set can contain up to 256 attributes. Attribute identifiers are encoded such that the most significant two nibbles specifies the attribute set and the least significant nibble specifies the attribute within the set. The currently defined attribute sets are listed in the following Table 8 Simple Metering Attribute Sets.
Table 8 Simple Metering Attribute Sets

<table>
<thead>
<tr>
<th>Attribute set identifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0</td>
<td>Summation Information Set</td>
</tr>
<tr>
<td>0x01</td>
<td>Meter Status</td>
</tr>
<tr>
<td>0x02</td>
<td>Consumption Formatting</td>
</tr>
<tr>
<td>0x03</td>
<td>ESP Historical Consumption</td>
</tr>
<tr>
<td>0x04</td>
<td>Load Profile Configuration</td>
</tr>
<tr>
<td>0x05 to 0xF</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

5.12.2.2.2.1 Summation Information Set

The following set of attributes provides the basic summed consumption values captured by the Energy, Gas, or Water metering device.

Table 9 Summation Information Attribute Set

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Name</th>
<th>Type</th>
<th>Range</th>
<th>Access</th>
<th>Default</th>
<th>Man. / Opt.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>CurrentSummationDelivered</td>
<td>Unsigned 40 bit Integer</td>
<td>0x000000000000 to 0xFFFFFFFFFF</td>
<td>Read Only</td>
<td>-</td>
<td>M</td>
</tr>
<tr>
<td>0x01</td>
<td>CurrentSummationReceived</td>
<td>Unsigned 40 bit Integer</td>
<td>0x000000000000 to 0xFFFFFFFFFF</td>
<td>Read Only</td>
<td>-</td>
<td>O</td>
</tr>
<tr>
<td>0x02</td>
<td>CurrentMaxDemandDelivered</td>
<td>Unsigned 40 bit Integer</td>
<td>0x000000000000 to 0xFFFFFFFFFF</td>
<td>Read Only</td>
<td>-</td>
<td>O</td>
</tr>
<tr>
<td>0x03</td>
<td>CurrentMaxDemandReceived</td>
<td>Unsigned 40 bit Integer</td>
<td>0x000000000000 to 0xFFFFFFFFFF</td>
<td>Read Only</td>
<td>-</td>
<td>O</td>
</tr>
<tr>
<td>0x04</td>
<td>CurrentTier1SummationDelivered</td>
<td>Unsigned 40 bit Integer</td>
<td>0x000000000000 to 0xFFFFFFFFFF</td>
<td>Read Only</td>
<td>-</td>
<td>O</td>
</tr>
<tr>
<td>0x05</td>
<td>CurrentTier1SummationReceived</td>
<td>Unsigned 40 bit Integer</td>
<td>0x000000000000 to 0xFFFFFFFFFF</td>
<td>Read Only</td>
<td>-</td>
<td>O</td>
</tr>
<tr>
<td>0x06</td>
<td>CurrentTier2SummationDelivered</td>
<td>Unsigned 40 bit Integer</td>
<td>0x000000000000 to 0xFFFFFFFFFF</td>
<td>Read Only</td>
<td>-</td>
<td>O</td>
</tr>
<tr>
<td>0x07</td>
<td>CurrentTier2SummationReceived</td>
<td>Unsigned 40 bit Integer</td>
<td>0x000000000000 to 0xFFFFFFFFFF</td>
<td>Read Only</td>
<td>-</td>
<td>O</td>
</tr>
<tr>
<td>0x08</td>
<td>CurrentTier3SummationDelivered</td>
<td>Unsigned 40 bit Integer</td>
<td>0x000000000000 to 0xFFFFFFFFFF</td>
<td>Read Only</td>
<td>-</td>
<td>O</td>
</tr>
<tr>
<td>0x09</td>
<td>CurrentTier3SummationReceived</td>
<td>Unsigned 40 bit Integer</td>
<td>0x000000000000 to 0xFFFFFFFFFF</td>
<td>Read Only</td>
<td>-</td>
<td>O</td>
</tr>
<tr>
<td>0x0A</td>
<td>CurrentTier4SummationDelivered</td>
<td>Unsigned 40 bit Integer</td>
<td>0x000000000000 to 0xFFFFFFFFFF</td>
<td>Read Only</td>
<td>-</td>
<td>O</td>
</tr>
<tr>
<td>0x0B</td>
<td>CurrentTier4SummationReceived</td>
<td>Unsigned 40 bit Integer</td>
<td>0x000000000000 to 0xFFFFFFFFFF</td>
<td>Read Only</td>
<td>-</td>
<td>O</td>
</tr>
<tr>
<td>0x0C</td>
<td>CurrentTier5SummationDelivered</td>
<td>Unsigned 40 bit Integer</td>
<td>0x000000000000 to 0xFFFFFFFFFF</td>
<td>Read Only</td>
<td>-</td>
<td>O</td>
</tr>
<tr>
<td>0x0D</td>
<td>CurrentTier5SummationReceived</td>
<td>Unsigned 40 bit Integer</td>
<td>0x000000000000 to 0xFFFFFFFFFF</td>
<td>Read Only</td>
<td>-</td>
<td>O</td>
</tr>
</tbody>
</table>
5.12.2.2.2.1.1 CurrentSummationDelivered Attribute

CurrentSummationDelivered represents the most recent summed value of Energy, Gas, or Water delivered and consumed in the premise. CurrentSummationDelivered is mandatory and must be provided as part of the minimum data set to be provided by the metering device. CurrentSummationDelivered is updated continuously as new measurements are made.

5.12.2.2.2.1.2 CurrentSummationReceived Attribute

CurrentSummationReceived represents the most recent summed value of Energy, Gas, or Water generated and delivered from the premise. If optionally provided, CurrentSummationReceived is updated continuously as new measurements are made.

5.12.2.2.2.1.3 CurrentMaxDemandDelivered Attribute

CurrentMaxDemandDelivered represents the maximum delivered value of Energy, Gas, or Water demand at the premise. If optionally provided, CurrentMaxDemandDelivered is updated continuously as new measurements are made.

5.12.2.2.2.1.4 CurrentMaxDemandReceived Attribute

CurrentMaxDemandReceived represents the maximum delivered value of Energy, Gas, or Water demand experienced at the premise. If optionally provided, CurrentMaxDemandReceived is updated continuously as new measurements are made.

5.12.2.2.2.1.5 CurrentTierNSummationDelivered Attributes

Attributes CurrentTier1SummationDelivered through CurrentTier5SummationDelivered represent the most recent summed value of Energy, Gas, or Water delivered to the premise during the appropriate Time of Use (TOU) periods. If optionally provided, attributes CurrentTier1SummationDelivered through CurrentTier5SummationDelivered are updated continuously as new measurements are made.

5.12.2.2.2.1.6 CurrentTierNSummationReceived Attributes

Attributes CurrentTier1SummationReceived through CurrentTier5SummationReceived represent the most recent summed value of Energy, Gas, or Water provided by the premise during the appropriate Time of Use (TOU) periods. If optionally provided, attributes CurrentTier1SummationReceived through CurrentTier5SummationReceived are updated continuously as new measurements are made.

5.12.2.2.2.1.7 DFTSummation Attribute

DFTSummation represents a snapshot of attribute CurrentSummationDelivered captured at the time indicated by attribute DailyFreezeTime. If optionally provided, DFTSummation is updated once every 24 hours and captured at the time set in section 5.12.2.2.2.1.8.

5.12.2.2.2.1.8 DailyFreezeTime Attribute

DailyFreezeTime represents the time of day when DFTSummation is captured. DailyFreezeTime is an unsigned 16 bit value representing the hour and minutes for DFT. The byte usages are:

- Bits 0 to 7: Range of 0 to 0x3C representing the number of minutes past the top of the hour.
- Bits 8 to 15: Range of 0 to 0x17 representing the hour of the day (in 24 hour format).
5.12.2.2.1.9 PowerFactor Attribute

*PowerFactor* contains the Power Factor ratio in 1/100’s. Valid values are 0 to 99.

### 5.12.2.3 Meter Status

#### Table 10 Meter Status Attribute Set

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Name</th>
<th>Type</th>
<th>Range</th>
<th>Access</th>
<th>Default Man. / Opt.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0</td>
<td>Status</td>
<td>Unsigned 8 bit Integer</td>
<td>0x00 to 0xFF</td>
<td>Read Only</td>
<td>0x00 M</td>
</tr>
<tr>
<td>0x1 to 0xFF</td>
<td>Reserved</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### 5.12.2.3.1 Status Attribute

*Status* attribute provides indicators reflecting the current error conditions found by the metering device. This attribute is an 8 bit field where when an individual bit is set, an error or warning condition exists. The behavior causing the setting or resetting each bit is device specific. The mapping of the bit is as follows:

<table>
<thead>
<tr>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reserved</td>
<td>Service Disconnect Engaged</td>
<td>Leak Detect</td>
<td>Power Quality</td>
<td>Power Failure</td>
<td>Tamper Detect</td>
<td>Low Battery</td>
<td>Check Meter</td>
</tr>
</tbody>
</table>

#### 5.12.2.4 Consumption Formatting

#### Table 11 Consumption Formatting Attribute Set

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Name</th>
<th>Type</th>
<th>Range</th>
<th>Access</th>
<th>Default Man. / Opt.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0</td>
<td>UnitofMeasure</td>
<td>8-bit Enumeration</td>
<td>0x00 to 0xFF</td>
<td>Read Only</td>
<td>0x00 M</td>
</tr>
<tr>
<td>0x1</td>
<td>Multiplier</td>
<td>Unsigned 24 bit Integer</td>
<td>0x00 to 0xFFFF</td>
<td>Read Only</td>
<td>- O</td>
</tr>
<tr>
<td>0x2</td>
<td>Divisor</td>
<td>Unsigned 24 bit Integer</td>
<td>0x0000000 to 0xFFFF</td>
<td>Read Only</td>
<td>- O</td>
</tr>
<tr>
<td>0x3</td>
<td>SummationFormatting</td>
<td>Unsigned 8 bit Integer</td>
<td>0x00 to 0xFF</td>
<td>Read Only</td>
<td>- M</td>
</tr>
<tr>
<td>0x4</td>
<td>DemandFormatting</td>
<td>Unsigned 8 bit Integer</td>
<td>0x00 to 0xFF</td>
<td>Read Only</td>
<td>- O</td>
</tr>
<tr>
<td>0x5</td>
<td>HistoricalConsumptionFormatting</td>
<td>Unsigned 8 bit Integer</td>
<td>0x00 to 0xFFFF</td>
<td>Read Only</td>
<td>- O</td>
</tr>
<tr>
<td>0x6 to 0xFF</td>
<td>Reserved</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### 5.12.2.4.1.1 UnitofMeasure Attribute

*UnitofMeasure* provides a label for the Energy, Gas, or Water being measured by the metering device. This Attribute is an 8 bit enumerated field representing the following:

- 0 = kWh (kilo-WattHours) & kW (kilo-Watts)
- 1 = m³ (Cubic Meter) & m³/h (Cubic Meter per Hour)
- 2 = ft³ (Cubic Feet) & ft³/h (Cubic Feet per Hour)
- 3 = ccf ((100 or Centum) Cubic Feet) & ccf/h ((100 or Centum) Cubic Feet per Hour)
- 4 = US gl (US Gallons) & US gl/h (US Gallons per Hour)
5 = IMP gl (Imperial Gallons) & IMP gl/h (Imperial Gallons per Hour)

5.12.2.2.4.1.2 Multiplier Attribute
Multiplier provides a value to be multiplied against a raw or uncompensated sensor count of Energy, Gas, or Water being measured by the metering device. If present, this attribute must be applied against all summed or consumption readings to derive the correct delivered and received values. This attribute must be used in conjunction with the Divisor Attribute.

5.12.2.2.4.1.3 Divisor Attribute
Divisor provides a value to divide the results of applying the Multiplier Attribute against a raw or uncompensated sensor count of Energy, Gas, or Water being measured by the metering device. If present, this attribute must be applied against all summed or consumption readings to derive the correct delivered and received values. This attribute must be used in conjunction with the Multiplier Attribute.

5.12.2.2.4.1.4 SummationFormatting Attribute
SummationFormatting provides a method to properly decipher the number of digits and the decimal location of the values found in the Summation Information Set of attributes. This attribute is to be decoded as follows:

| Bits 0 to 2: Number of Digits to the right of the Decimal Point
| Bits 3 to 6: Number of Digits to the left of the Decimal Point
| Bit 7: If set, suppress leading zeros.

This attribute shall be used against the following attributes:
- CurrentSummationDelivered
- CurrentSummationReceived
- CurrentTier1SummationDelivered
- CurrentTier1SummationReceived
- CurrentTier2SummationDelivered
- CurrentTier2SummationReceived
- CurrentTier3SummationDelivered
- CurrentTier3SummationReceived
- CurrentTier4SummationDelivered
- CurrentTier4SummationReceived

5.12.2.2.4.1.5 DemandFormatting Attribute
DemandFormatting provides a method to properly decipher the number of digits and the decimal location of the values found in the Demand related attributes. This attribute is to be decoded as follows:

| Bits 0 to 2: Number of Digits to the right of the Decimal Point
| Bits 3 to 6: Number of Digits to the left of the Decimal Point
| Bit 7: If set, suppress leading zeros.

This attribute shall be used against the following attributes:
- CurrentMaxDemandDelivered
- CurrentMaxDemandReceived
- CurrentDemand

5.12.2.2.4.1.6 HistoricalConsumptionFormatting Attribute
HistoricalConsumptionFormatting provides a method to properly decipher the number of digits and the decimal location of the values found in the ESP Historical Consumption Set of attributes. This attribute is to be decoded as follows:

| Bits 0 to 2: Number of Digits to the right of the Decimal Point
| Bits 3 to 6: Number of Digits to the left of the Decimal Point
| Bit 7: If set, suppress leading zeros.
This attribute shall be used against the following attributes:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Range</th>
<th>Access</th>
<th>Default</th>
<th>Man. / Opt.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CurrentDayConsumptionDelivered</td>
<td>Unsigned 24 bit Integer</td>
<td>0x00000000 to 0xFFFFFFFF</td>
<td>Read Only</td>
<td>- 0</td>
<td></td>
</tr>
<tr>
<td>CurrentDayConsumptionReceived</td>
<td>Unsigned 24 bit Integer</td>
<td>0x00000000 to 0xFFFFFFFF</td>
<td>Read Only</td>
<td>- 0</td>
<td></td>
</tr>
<tr>
<td>PreviousDayConsumptionDelivered</td>
<td>Unsigned 24 bit Integer</td>
<td>0x00000000 to 0xFFFFFFFF</td>
<td>Read Only</td>
<td>- 0</td>
<td></td>
</tr>
<tr>
<td>PreviousDayConsumptionReceived</td>
<td>Unsigned 24 bit Integer</td>
<td>0x00000000 to 0xFFFFFFFF</td>
<td>Read Only</td>
<td>- 0</td>
<td></td>
</tr>
</tbody>
</table>

5.12.2.2.4.2 ESP Historical Consumption

Table 12 ESP Historical Attribute Set

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Name</th>
<th>Type</th>
<th>Range</th>
<th>Access</th>
<th>Default</th>
<th>Man. / Opt.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0</td>
<td>InstantaneousDemand</td>
<td>Unsigned 24 bit Integer</td>
<td>0x00000000 to 0xFFFFFFFF</td>
<td>Read Only</td>
<td>0x00</td>
<td></td>
</tr>
<tr>
<td>0x1</td>
<td>CurrentDayConsumptionDelivered</td>
<td>Unsigned 24 bit Integer</td>
<td>0x00000000 to 0xFFFFFFFF</td>
<td>Read Only</td>
<td>- 0</td>
<td></td>
</tr>
<tr>
<td>0x2</td>
<td>CurrentDayConsumptionReceived</td>
<td>Unsigned 24 bit Integer</td>
<td>0x00000000 to 0xFFFFFFFF</td>
<td>Read Only</td>
<td>- 0</td>
<td></td>
</tr>
<tr>
<td>0x3</td>
<td>PreviousDayConsumptionDelivered</td>
<td>Unsigned 24 bit Integer</td>
<td>0x00000000 to 0xFFFFFFFF</td>
<td>Read Only</td>
<td>- 0</td>
<td></td>
</tr>
<tr>
<td>0x4</td>
<td>PreviousDayConsumptionReceived</td>
<td>Unsigned 24 bit Integer</td>
<td>0x00000000 to 0xFFFFFFFF</td>
<td>Read Only</td>
<td>- 0</td>
<td></td>
</tr>
<tr>
<td>0x05</td>
<td>CurrentPartialProfileIntervalStartTime</td>
<td>UTCTime</td>
<td></td>
<td>Read Only</td>
<td>- 0</td>
<td></td>
</tr>
<tr>
<td>0x06</td>
<td>CurrentPartialProfileIntervalValue</td>
<td>Unsigned 24 bit Integer</td>
<td>0x00000000 to 0xFFFFFFFF</td>
<td>Read Only</td>
<td>- 0</td>
<td></td>
</tr>
<tr>
<td>0x7 to 0xFF</td>
<td>Reserved</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5.12.2.2.4.2.1 InstantaneousDemand Attribute

*InstantaneousDemand* represents the current Demand of Energy, Gas, or Water delivered at the premise. *InstantaneousDemand* is updated continuously as new measurements are made. The frequency of updates to this field is specific to the metering device, but should be limited to a maximum of once per second.

5.12.2.2.4.2.2 CurrentDayConsumptionDelivered Attribute

*CurrentDayConsumptionDelivered* represents the summed value of Energy, Gas, or Water generated and delivered to the premise since midnight UTC. If optionally provided, *CurrentDayConsumptionDelivered* is updated continuously as new measurements are made.

5.12.2.2.4.2.2.1 CurrentDayConsumptionReceived Attribute

*CurrentDayConsumptionReceived* represents the summed value of Energy, Gas, or Water generated and received from the premise since midnight UTC. If optionally provided, *CurrentDayConsumptionReceived* is updated continuously as new measurements are made.

5.12.2.2.4.2.3 PreviousDayConsumptionDelivered Attribute

*PreviousDayConsumptionDelivered* represents the summed value of Energy, Gas, or Water generated and delivered to the premise within the previous 24 hour period starting at midnight UTC. If optionally provided, *PreviousDayConsumptionDelivered* is updated every midnight UTC.

5.12.2.2.4.2.4 PreviousDayConsumptionReceived Attribute

*PreviousDayConsumptionReceived* represents the summed value of Energy, Gas, or Water generated and received from the premise within the previous 24 hour period starting at midnight UTC. If optionally provided, *PreviousDayConsumptionReceived* is updated every midnight UTC.
5.12.2.4.2.5 CurrentPartialProfileIntervalStartTime

*CurrentPartialProfileIntervalStartTime* represents the start time of the current Load Profile interval being accumulated.

5.12.2.4.2.6 CurrentPartialProfileIntervalValue

*CurrentPartialProfileIntervalValue* represents the value of the current Load Profile interval being accumulated.

### 5.12.2.4.3 Load Profile Configuration

**Table 13 Load Profile Configuration Attribute Set**

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Name</th>
<th>Type</th>
<th>Range</th>
<th>Access</th>
<th>Default</th>
<th>Man. / Opt.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0</td>
<td>ProfileIntervalPeriod</td>
<td>8-bit Enumeration</td>
<td>0x00 to 0xFF</td>
<td>Read Only</td>
<td>0x00</td>
<td>O</td>
</tr>
<tr>
<td>0x1</td>
<td>NumberOfPeriodsDelivered</td>
<td>Unsigned 8 bit Integer</td>
<td>0x00 to 0xFF</td>
<td>Read Only</td>
<td>0x18</td>
<td>O</td>
</tr>
<tr>
<td>0x2 to 0xFF</td>
<td>Reserved</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5.12.2.4.3.1 ProfileIntervalPeriod Attribute

*ProfileIntervalPeriod* represents the interval or time frame used to capture metered Energy, Gas, and Water consumption for profiling purposes. If optionally provided, *ProfileIntervalPeriod* is an enumerated field representing the following timeframes:

- 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

5.12.2.4.3.2 NumberOfPeriodsDelivered Attribute

*NumberOfPeriodsDelivered* represents the number of intervals returned by the Get Profile Response command. Please note the number of periods returned in the Get Profile Response command can be calculated when the packets are received and can replace the usage of this attribute. The intent is to provide this information as a convenience in predetermining how often to use the Get Profile Command.

### 5.12.2.5 Server Commands

#### 5.12.2.5.1 Commands Generated

The command IDs generated by the Simple Metering server cluster are listed in Table 14.
Table 14 – Generated Command IDs for the Simple Metering server

<table>
<thead>
<tr>
<th>Command identifier field value</th>
<th>Description</th>
<th>Mandatory / Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>Get Profile Response</td>
<td>O</td>
</tr>
<tr>
<td>0x01 – 0xff</td>
<td>Reserved</td>
<td></td>
</tr>
</tbody>
</table>

5.12.2.2.5.2 Get Profile Response Command

5.12.2.2.5.2.1 Payload Format

The Get Profile Response command payload shall be formatted as illustrated in Figure 13 Get Profile Response Command Payload.

**Figure 13 Get Profile Response Command Payload**

<table>
<thead>
<tr>
<th>Bits</th>
<th>32</th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Type</td>
<td>UTC Time</td>
<td>Array of Unsigned 24 bit Integers</td>
</tr>
<tr>
<td>Field Name</td>
<td>EndTime</td>
<td>Intervals</td>
</tr>
</tbody>
</table>

5.12.2.2.5.2.2 Payload Details

**EndTime:** 32 bit value (in UTC) representing the end time of the most chronologically recent interval being requested. Example: Data collected from 2:00 PM to 3:00 PM would be specified as a 3:00 PM interval (end time)

**Intervals:** 32 bit value (in UTC) used to select an Intervals block from all the Intervals blocks available. The EndTime should be equal to the End Time of the most recent interval of the block requested. The most recent Intervals block is requested using an End Time set to 0xFFFFFFFF, subsequent Intervals block are requested using an End time set to the EndTime of the previous block - (number of intervals of the previous block * ProfileIntervalPeriod). Intervals are represented in the same Unit Of Measure as as attribute *CurrentSummationDelivered*. Number of intervals returned is specified by the Attribute *NumberOfPeriodsDelivered*. Data is organized in a reverse chronological order. The most recent interval is transmitted first and the oldest interval is transmitted last. Invalid intervals or non-captured intervals should be marked as 0xFFFFFFFF.

5.12.2.2.5.2.3 When generated

This command is generated when the Client command GetProfile is received.

5.12.2.2.5.2.4 Attribute reporting

Not used.
5.12.2.3 Client Commands

5.12.2.3.1 Get Profile Command

The Get Profile command payload shall be formatted as illustrated in Table 15.

<table>
<thead>
<tr>
<th>Table 15 GetProfile Payload</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bits</td>
</tr>
<tr>
<td>Data Type</td>
</tr>
<tr>
<td>Field Name</td>
</tr>
</tbody>
</table>

5.12.2.3.1.1 Payload Details

Channel: Enumerated value representing the return of Delivered or Received Intervals where:

- 0 = Delivered
- 1 = Received

EndTime: 32 bit value representing the starting Time (in UTC) of intervals to be returned. For convenience, a UTCTime of 0xFFFFFFFF would indicate to return the most recent block of Load Profile Data.

5.12.2.3.1.2 When generated

The GetProfile command is generated when a client device wishes to retrieve a list of captured Energy, Gas or water consumption for profiling purposes.

5.12.2.3.1.3 Command Processing Response

If success or failure occurs in recognizing or processing the payload of the GetProfile command, the appropriate enumerated ZCL status (as referenced in the ZCL Cluster Library specification) will be returned.

5.12.2.3.1.4 Attribute reporting

Not used.

5.12.3 Price Cluster

5.12.3.1 Overview

Attributes and commands for sending commodity pricing information.
5.12.3.2 Server

5.12.3.2.1 Dependencies

None.

5.12.3.2.2 Attributes

None.

5.12.3.2.3 Commands Received

The server side of the Price cluster is capable of receiving the commands listed in Table 16.

Table 16 Received Command IDs for the Price cluster

<table>
<thead>
<tr>
<th>Command identifier field value</th>
<th>Description</th>
<th>Mandatory / Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>Get Current Price</td>
<td>M</td>
</tr>
<tr>
<td>0x01</td>
<td>Get Scheduled Prices</td>
<td>O</td>
</tr>
<tr>
<td>0x02 – 0xff</td>
<td>Reserved</td>
<td></td>
</tr>
</tbody>
</table>

5.12.3.2.3.1 Get Current Price Command

This command initiates a Publish Price command (see 5.12.3.2.4.1) for the current time.

5.12.3.2.3.1.1 Payload Format

The payload of the Get Current Price command is formatted as shown in Figure 14.

![Figure 14 The Format of the Get Current Price command payload](image)

5.12.3.2.3.1.2 Payload Details

**The Command Options Field:** The command options field is a single octet in length and is formatted as a bit-field, as follows:

<table>
<thead>
<tr>
<th>Bits</th>
<th>0</th>
<th>1…7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field Name</td>
<td>Requestor Rx On When Idle</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

**The Requestor Rx On When Idle sub-field:** The Requestor Rx On When Idle sub-field has a value of 1 if the requestor’s receiver may be, for all practical purposes, enabled when the device is not actively transmitting, thereby making it very likely that regular broadcasts of pricing information will be received by this device, and 0 otherwise.

A device that publishes price information may use the value of this bit, as received from requestors in its neighborhood, to determine publishing policy. For example, if a device makes a request for current pricing information and the requestor Rx on when idle sub-field of the GetCurrentPrice command payload has a value of 1, indicating that the device will be likely to receive regular price messages,
then the receiving device may store information about the requestor and use it in future publishing operations.

5.12.3.2.3.1.3 Effect on Receipt
On receipt of this command, the device shall send a Publish Price command (see 5.12.3.2.4.1) with the information for the current time.

5.12.3.2.3.2 Get Scheduled Prices Command
This command initiates a Publish Price command (see 5.12.3.2.4.1) for all currently scheduled times. A server device shall be capable of storing five scheduled price events at a minimum.

5.12.3.2.3.2.1 Payload Details
This command has no payload.

5.12.3.2.3.2.2 Effect on Receipt
On receipt of this command, the device shall send a Publish Price command (see 5.12.3.2.4.1) for all currently scheduled price events.

5.12.3.2.4 Commands Generated
The server side of the Price cluster is capable of generating the commands listed in Table 17.

<table>
<thead>
<tr>
<th>Command identifier field value</th>
<th>Description</th>
<th>Mandatory / Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>Publish Price</td>
<td>M</td>
</tr>
<tr>
<td>0x01 – 0xff</td>
<td>Reserved</td>
<td></td>
</tr>
</tbody>
</table>

5.12.3.2.4.1 Publish Price Command
The Publish Price command is generated in response to receiving a Get Current Price command (see 5.11.3.2.3.1), a Get Scheduled Prices command (see 5.11.3.2.3.2), or when an update to the pricing information is available from the commodity provider. When a Get Current Price or Get Scheduled Prices command is received over a ZigBee AMI network, the Publish Price command should be 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 AMI network. When responding to a request via the Inter-PAN SAP, the Publish Price command should be broadcast to the PAN of the requester after a delay of 0.5 seconds, to avoid a potential broadcast storm of packets.

Nested and overlapping Publish Price commands are not allowed. The current active price will be replaced if new price information is received by the ESP.

5.12.3.2.4.1.1 Payload Format
The PublishPrice command payload shall be formatted as illustrated in Figure 15.

<table>
<thead>
<tr>
<th>Octets</th>
<th>4</th>
<th>0 – 8 characters</th>
<th>4</th>
<th>1</th>
<th>2</th>
<th>1</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Type</td>
<td>Unsized 32 bit</td>
<td>Character String</td>
<td>UTCTime</td>
<td>8 bits enumeration</td>
<td>Unsigned 16 bit</td>
<td>Unsigned 8 bit</td>
<td>UTCTime</td>
</tr>
<tr>
<td>Field Name</td>
<td>Provider ID</td>
<td>Rate Label</td>
<td>Current Time</td>
<td>Unit of Measure</td>
<td>Currency</td>
<td>Price Trailing Digit</td>
<td>Start Time</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
<td>------------</td>
<td>--------------</td>
<td>-----------------</td>
<td>----------</td>
<td>----------------------</td>
<td>------------</td>
</tr>
<tr>
<td></td>
<td>Integer</td>
<td>Integer</td>
<td>Integer</td>
<td>Integer</td>
<td>Integer</td>
<td>Integer</td>
<td>Integer</td>
</tr>
</tbody>
</table>

Continuation of payload...

<table>
<thead>
<tr>
<th>2</th>
<th>4</th>
<th>1</th>
<th>4</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Duration In Seconds</th>
<th>Price</th>
<th>Price Ratio</th>
<th>Generation Price</th>
<th>Generation Price Ratio</th>
<th>Price Tier</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integer</td>
<td>Integer</td>
<td>Integer</td>
<td>Integer</td>
<td>Integer</td>
<td>Integer</td>
</tr>
</tbody>
</table>

**Figure 15 Format of the Publish Price Command payload**

**Provider ID:** An unsigned 32 bit field containing a unique identifier for the commodity provider. This field is thought to be useful in deregulated markets where multiple commodity providers may be available.

**Rate Label:** A character String field containing commodity provider-specific information regarding the current billing rate. This field is thought to be useful when a commodity provider may have multiple pricing plans.

**Current Time:** A UTCTime field containing the current time as determined by the device. This field is thought to be useful to provide an extra value-added feature for the broadcast price signals.

**Unit of Measure:** An 8 bit enumeration field identifying the commodity as well as its base unit of measure. The enumeration used for this field shall match that of the Simple Metering cluster (see 5.12.2.2.4.1.1).

**Currency:** An unsigned 32 bit field containing identifying information concerning the local unit of currency used in the price field. This field is thought to be useful for displaying the appropriate symbol for a currency (i.e.: $).

The value of the currency field should match the values defined by ISO 4127.

**Price Trailing Digit:** An unsigned 8 bit field used to determine where the decimal point is located in the price field. The value of the Price Trailing Digit field indicates the number of digits to the right of the decimal point.

**Start Time:** A UTCTime field to denote the time at which the price signal becomes valid. A start time of 0xffffffff is a special time denoting “now.”

**Duration In Seconds:** An unsigned 16 bit field used to denote the amount of time in seconds after the Start Time during which the price signal is valid. Maximum value means “forever” duration.

**Price:** 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 entering the premise.

**Price Ratio:** 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.

**Generation Price:** 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 exiting the premise. 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.
**Generation Price Ratio**: 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.

**Price Tier**: An unsigned 8 bit field indicating the current pricing tier as chosen by the commodity provider. Example tiers include shoulder, normal, peak, and critical peak.

5.12.3.2.4.2 Effect on Receipt

On receipt of this command, the device is informed of a price event for the specific provider, commodity, and currency indicated.

5.12.3.3 Client

5.12.3.3.1 Dependencies

None.

5.12.3.3.2 Attributes

None.

5.12.3.3.3 Commands Received

The client receives the cluster specific response commands detailed in 5.12.3.2.3.1

5.12.3.3.4 Commands Generated

The client generates the cluster specific commands detailed in 5.12.3.2.4.1, as required by the application.

5.12.4 Messaging Cluster

5.12.4.1 Overview

This cluster provides an interface for passing text messages between AMI devices. Messages are expected to be delivered via the ESP and then optionally delivered immediately onto the appropriate AMI devices, or just made available to all AMI devices for later pickup. Nested and overlapping messages are not allowed. The current active message will be replaced if a new message is received by the ESP.
5.12.4.2 Server

5.12.4.2.1 Dependencies
Support for ZCL Data Types.
Anonymous Inter-PAN transmission mechanism outlined in Annex Error! Reference source not found.

5.12.4.2.2 Attributes
None.

5.12.4.2.3 Commands Generated
The command IDs generated by the Messaging server cluster are listed in Table 18.

<table>
<thead>
<tr>
<th>Command identifier field value</th>
<th>Description</th>
<th>Mandatory/Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>Display Message</td>
<td>M</td>
</tr>
<tr>
<td>0x01</td>
<td>Cancel Message</td>
<td>M</td>
</tr>
<tr>
<td>0x02 – 0xff</td>
<td>Reserved</td>
<td></td>
</tr>
</tbody>
</table>

5.12.4.2.3.1 Display Message( )

5.12.4.2.3.1.1 Payload Format
The Display Message command payload shall be formatted as illustrated in Table 19 Display Message command payload.
Table 19 Display Message command payload

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Bits</th>
<th>32</th>
<th>8</th>
<th>8</th>
<th>16</th>
<th>Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message ID</td>
<td>Message ID</td>
<td>Message Control</td>
<td>Message Urgency</td>
<td>Duration In Seconds</td>
<td>Message</td>
<td></td>
</tr>
<tr>
<td>Message Type</td>
<td>Unsigned 32-bit integer</td>
<td>Unsigned 8-bit integer</td>
<td>Unsigned 8-bit integer</td>
<td>Unsigned 16 bit Integer</td>
<td>Character string</td>
<td></td>
</tr>
</tbody>
</table>

5.12.4.2.3.1.2 Payload Details

**Message ID:** 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.

**MessageControl:** An enumerated field indicating the need to optionally pass the message onto the Anonymous Inter-PAN transmission mechanism outlined in Annex B or that a user confirmation is required for a message. Bit encoding of this field is:

- Bits 0 to 6: Enumerated values of:
  - 0 = Send message through normal command function to client.
  - 1 = Additionally pass message onto the Anonymous Inter-PAN transmission mechanism.
  - 2 = Only send message through the Anonymous Inter-PAN transmission mechanism.
  - 3 to 127 = Reserved.

- Bit 7: Message Confirmation where:
  - 0 = Not Required
  - 1 = Required

If the Anonymous Inter-PAN transmission mechanism outlined in Annex B is not supported on a particular device, Bits 0 to 6 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’s either confirmed via a button or by selecting a confirmation option on the device. Confirmation is typically used when the Utility is sending down information such as a disconnection notice, or prepaid billing information. Message duration is ignored when confirmation is requested and the message is displayed until confirmed.

Note: It is desired that the device provide a visual indicator (flashing display or indicate with its LED’s as examples) that a message requiring confirmation is being displayed, and requires confirmation.

**MessageUrgency:** An enumerated field indicating the importance level of the message to be transferred. The enumerated values for this field are:

- 0 = Low
- 1 = Medium
- 3 = High
- 4 = Critical

**Duration In Seconds:** An unsigned 16 bit field is used to denote the amount of time in seconds after the Start Time during which the message is valid. Maximum value means “forever” duration.
Message: A ZCL String containing the message to be delivered. The String shall be encoded in the UTF-8 format. Please note: Since the Anonymous Inter-PAN transmission mechanism outlined in Annex Error! Reference source not found. does not support fragmentation it is limited in its message size, any message forwarded will be truncated to match the maximum message length supported.

5.12.4.2.3.2 Cancel Message( )

The CancelMessage command provides the ability to cancel the sending or acceptance of previously sent messages. When this message is received the recipient device has the option of clearing any display or user interfaces it supports, or has the option of logging the message for future reference.

Table 20 Cancel Message command payload

<table>
<thead>
<tr>
<th>bits</th>
<th>32</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field Name</td>
<td>Message ID</td>
<td>Message Control</td>
</tr>
<tr>
<td>Data Type</td>
<td>Unsigned 32-bit integer</td>
<td>Unsigned 8-bit integer</td>
</tr>
</tbody>
</table>

5.12.4.2.3.2.1 Payload Details

Message ID: A unique unsigned 32 bit number identifier for the message being cancelled. It’s expected the value contained in this field is a unique number managed by upstream systems or a UTC based time stamp (UTCTime data type) identifying when the message was originally issued.

MessageControl: An enumerated field that indicating the optional ability to pass the cancel message request onto the Anonymous Inter-PAN transmission mechanism outlined in Annex Error! Reference source not found. If the Anonymous Inter-PAN transmission mechanism is not supported on a particular device, this parameter is ignored.

Enumerated values are:
0 = Send message cancellation through normal command functions.
1 = Additionally pass message cancellation onto the Anonymous Inter-PAN transmission mechanism.
2 = Only send message cancellation through the Anonymous Inter-PAN transmission mechanism.
3 to 255 = Reserved.

5.12.4.3 Client

5.12.4.3.1 Dependencies

Support for ZCL Data Types.

5.12.4.3.2 Attributes

None.
5.12.4.3.3 Commands Generated

The command IDs generated by the Messaging cluster are listed in Table 21 Messaging Client Commands.

<table>
<thead>
<tr>
<th>Command identifier</th>
<th>Description</th>
<th>Mandatory / Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>Get Last Message( )</td>
<td>M</td>
</tr>
<tr>
<td>0x01</td>
<td>Message Confirmation()</td>
<td>M</td>
</tr>
<tr>
<td>0x02 – 0xff</td>
<td>Reserved</td>
<td></td>
</tr>
</tbody>
</table>

5.12.4.3.3.1 Get Last Message( )

This command has no payload.

5.12.4.3.3.1.1 Effect on Receipt

On receipt of this command, the device shall send a Display Message command (refer to 5.12.4.2.3.1).

5.12.4.3.3.2 Message Confirmation( )

The Message Confirmation command provides the ability to cancel the sending or acceptance of previously sent messages.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Data Type</th>
<th>Bits</th>
<th>32</th>
<th>32</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message ID</td>
<td>Unsigned 32-bit integer</td>
<td>UTCTime</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Confirmation Time</td>
<td>UTCTime</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5.12.4.3.3.2.1 Payload Details

Message ID: A unique unsigned 32 bit number identifier for the message being cancelled.

Confirmation Time: UTCTime of user confirmation of message.

5.12.4.4 Application Guidelines

For Server and Client transactions, please refer to Figure 17.
5.12.5 Registration Cluster

5.12.5.1 Overview

This cluster provides an interface to control the registration process. This allows a simple set of controls and indicators allowing the client to request a device to complete the AMI authentication process, discard its AMI authentication information, or provide its AMI authentication status. Note: the registration cluster does not provide any authentication or authorization services, it merely provides a uniform set of controls to trigger the AMI authentication process.

5.12.5.2 Server

Any device that can be registered functions as a registration server. This means that In Home Displays, Programmable Thermostats, Energy Management Consoles etc., would be registration servers.

5.12.5.2.1 Dependencies

Support for ZCL Data Types.
Support for Inter-PAN Communication.

5.12.5.2.2 Registration Cluster Attributes

The following set of attributes provides the basic registration status information. Note: These attributes are expected to be updated whenever the state changes. These attributes must be updated when the appropriate commands in this cluster are executed. For example, when a device receives anonymous data, it should update the state of the AnonymousDataAvailable attribute. When a device is specifically asked to verify whether anonymous data is available, it must update the attribute.

Table 22 Registration Cluster Attributes

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Name</th>
<th>Type</th>
<th>Range</th>
<th>Access</th>
<th>Default</th>
<th>M./O.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0</td>
<td>RegistrationState</td>
<td>Unsigned 8 bit Integer</td>
<td>0x00 to 0xFF</td>
<td>Read Only</td>
<td>-</td>
<td>M</td>
</tr>
<tr>
<td>0x1</td>
<td>AnonymousDataAvailable</td>
<td>Unsigned 8 bit Integer</td>
<td>0x00 to 0xFF</td>
<td>Read Only</td>
<td>-</td>
<td>M</td>
</tr>
<tr>
<td>0x2</td>
<td>RegisteredClientAddress</td>
<td>IEEE Address</td>
<td>0x00 to 0xFFFF</td>
<td>Read Only</td>
<td>-</td>
<td>M</td>
</tr>
<tr>
<td>30x3 to 0xF</td>
<td>Reserved</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5.12.5.2.2.1 RegistrationState Attribute

The RegistrationState represents the current registration state. This is an enumerated value. Valid states are:

- 0 = Unregistered
- 1 = Joining Network
- 2 = Joined Network
- 3 = Submitted Registration Request
- 4 = Registration Rejected
- 5 = Registered
- 6 = Registration Not Possible

All other possible values are currently reserved, but may be used in subsequent versions of the profile.

5.12.5.2.2.2 AnonymousDataAvailable Attribute

The AnonymousDataAvailable attribute is set if the device is able to contact a provider of anonymous data. This is an enumerated value. Valid states are:

- 0 = No source for anonymous data has been found.
- 1 = A source for anonymous data has been found.

All other possible values are currently reserved, but may be used in subsequent versions of the profile.

5.12.5.2.2.3 RegisteredClientAddress Attribute

5.12.5.2.3 The RegisteredClientAddress attribute stores the IEEE Address of the client device managing registration. Commands Generated

The command IDs generated by the Registration server cluster are listed in Table 23.
Table 23 – Generated Command IDs for the Registration server Cluster

<table>
<thead>
<tr>
<th>Command identifier field value</th>
<th>Description</th>
<th>Mandatory / Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>Initiate Registration()</td>
<td>M</td>
</tr>
<tr>
<td>0x01</td>
<td>Initiate DeRegistration()</td>
<td>M</td>
</tr>
<tr>
<td>0x02</td>
<td>Scan For Anonymous Data</td>
<td>M</td>
</tr>
<tr>
<td>0x03 – 0xff</td>
<td>Reserved</td>
<td></td>
</tr>
</tbody>
</table>

5.12.5.2.3.1 Initiate Registration()

5.12.5.2.3.1.1 Payload Details
This command has no payload.

5.12.5.2.3.1.2 Effect on Receipt
Upon receiving this command, the device will initiate the registration process. This means attempting to authenticate via the Key Establishment Cluster.

5.12.5.2.3.2 Initiate DeRegistration()

5.12.5.2.3.2.1 Payload Details
This command has no payload.

5.12.5.2.3.2.2 Effect on Receipt
Upon receiving this command, the device will initiate the deregistration process. This means attempting to notify the server it authenticated against, and then revoking any current authorizations. Unless dual profiles are supported, once completed, the device should leave the ZigBee AMI network.

5.12.5.2.3.3 Scan For Anonymous Data()

5.12.5.2.3.3.1 Payload Details
This command has no payload.

5.12.5.2.3.3.2 Effect on Receipt
Upon receiving this command, the device will use the anonymous data cluster to send out requests for pricing and/or messaging data. If the device receives properly formed responses within 5 seconds, it will update the AnonymousDataAvailable attribute to indicate that anonymous data is available. Otherwise it will update the attribute to indicate it is not available.

5.12.5.3 Client
The ESP is the client of the registration cluster. At any point it may request that a registered device reregister or deregister. It also keeps track of which devices have registered through it, and exposes this through a set of attributes.
5.12.5.3.1 Dependencies

Support for ZCL Data Types.

5.12.5.3.2 Attributes

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Name</th>
<th>Type</th>
<th>Range</th>
<th>Access</th>
<th>Default</th>
<th>Man. / Opt.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0</td>
<td>MaxDevices</td>
<td>Unsigned 8 bit Integer</td>
<td>0x00 to 0xFF</td>
<td>Read Only</td>
<td>-</td>
<td>M</td>
</tr>
<tr>
<td>0x1</td>
<td>DevicesRegistered</td>
<td>Unsigned 8 bit Integer</td>
<td>0x00 to 0xFF</td>
<td>Read Only</td>
<td>-</td>
<td>M</td>
</tr>
<tr>
<td>0x2</td>
<td>DeviceIDArray</td>
<td>Array of IEEE Addresses</td>
<td>0 to MaxDevices</td>
<td>Read Only</td>
<td>-</td>
<td>M</td>
</tr>
<tr>
<td>0x5 to 0xF</td>
<td>Reserved</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5.12.5.3.2.1 MaxDevices Attribute

The `MaxDevices` attribute represents the maximum number of devices that can register with this device.

5.12.5.3.2.2 DevicesRegistered Attribute

The `DevicesRegistered` attribute represents the current count of devices that are registered with this ESP. This value can’t exceed the value contained in the `MaxDevices` attribute. If no devices are registered, this attribute is 0x00.

5.12.5.3.2.3 DeviceIDArray Attribute

The `DeviceIDArray` attribute represents an Array of IEEE Addresses containing the EUI64 addresses of the devices registered with this device. If no devices are registered, this array is empty.

5.12.5.3 Commands Received

No client commands are supported.

5.13 Commissioning

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

The ZigBee Alliance has recognized the importance of commissioning and, in particular, the importance of specifications for network and stack commissioning in a multi-vendor environment. Thus, network and stack commissioning procedures are being designed outside the context of any particular profile, where possible, and grouped under the auspices of the Commissioning Tools Task Group (CTTG). This task group is developing a commissioning framework specification [R3].
5.13.1 Forming the network (Start-up sequence)

AMI devices must form their own network or join an existing network. The commissioning framework [R3] discusses some of the relevant issues in this procedure.

It is intended that an installer of an AMI device know if the device is forming a network or joining an existing network.

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

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

5.13.2 Support for commissioning modes

Three different commissioning modes are discussed in [R4]. They are denoted A, E and S-mode.

As discussed above, AMI devices will either automatically form or join a network based on preinstalled information such as extended PANID, or will join an existing network that has permit joining on.

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

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

5.13.3 Commissioning Documentation

To ensure a uniform user experience when commissioning AMI devices, all ZigBee AMI devices are required to provide documentation with their product that explains how to perform device commissioning in using a common language set, i.e., “form network”, “join network”, etc. The following table is representative of what should be included in the product documentation and also shows the common language to be used. Note that some items are not required to be supported by the device, but if not supported, should be indicated as such in the table. Using a common language set among all AMI devices will make it intuitive for a user or installer to start or join a network, even if the new device is manufactured by a different OEM than the devices in an existing network.

Example Commissioning Documentation

<table>
<thead>
<tr>
<th>ZigBee Action</th>
<th>What to do to device to perform action</th>
</tr>
</thead>
</table>

Copyright © 2007, The ZigBee Alliance. All rights reserved.
Join Network | Press and hold the up and down buttons
Form Network | Start the device
Operate unregistered | Start the device
Allow Others To Join Network | Hold the red button and press the green button four times
Restore Factory Fresh Settings | Hold the red and green buttons down simultaneously for 15 seconds
Install Start Up Parameters | Use optical interface and manufacturers tool

How to program preconfigured security key

The [R4] chapter 7 lists what commissioning actions are mandatory or optional on each device. The following section describes each action:

“Join Network” Go find and join first available network.

“Form Network” For devices that can start a network.

“Allow Others to Join Network” For routers and coordinators only. Allows you to add more nodes to an existing network. This must have a mandatory maximum timeout of 4.25 minutes.

“Restore to Factory Fresh Settings” – Restore the device settings to fresh state. (also performs leave).

“Install Start Up Parameters” – Document the method used to install the start up parameters sufficient to allow a device to start a new network or join an existing network. This may involve the use of a manufacturers specific tool or may be a documented over the air or out of band interface.

If a Device does not support an action it must be listed in its documentation as “Not Supported”.

The default permit joining time will be 3 minutes. Please note the “mgmt_permit_joining_request” has a maximum value of 4.25 minutes.

### 5.13.4 Commissioning Procedure for Different Network Types

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

#### 5.13.4.1 Commissioning for Neighborhood Area Network or Sub-metering

Under a neighborhood area network, other meters such as gas or water meters may join electric meters that form a backbone of the network. The process of joining the network is separate from the process for device binding where the device billing information is configured for a particular dwelling unit. It may be desirable to allow the meter to join an adjacent dwelling unit from a network standpoint to ensure proper connectivity. The application level will handle the configuration of the billing information later.
1. There are two methods for joining such a device onto an existing network:
   a. The device is commissioned using a tool with the necessary network and security
      start up parameters to allow it to rejoin the network as a new device. The device can
      rejoin any device in the network since it has all the network information.
   b. The network has permit joining turned on by an external tool and the device joins this
      network and undergoes joining and authentication as any newly joined device.

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

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

5.13.4.2 Commissioning for Home Area Network

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

1. The AMI 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).

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

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

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

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

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

5.13.4.3 Network Startup and Certificate Exchange Diagram

The following diagram depicts an example of a successful network startup and certificate exchange
(without pre-established link keys). Please refer to Annex C for further discussions on communication
exchanges and key support.
Figure 19 Successful network startup and certificate exchange
5.13.5 AMI Tunneling (Complex Metering) Cluster

TBD in a future revision of the AMI Profile specification.

5.13.6 Pre-Payment Cluster

TBD in a future revision of the AMI Profile specification.
6 Constants, error codes and general alarms

Profile-specific constants are shown in Table 24.

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
7 Device Specifications

7.1 Common Clusters

Support for certain clusters is common to all the devices in this profile. The clusters shown in Table 25 shall be supported by all devices in this profile as mandatory or optional according the designation given here. Individual device descriptions may place further restrictions on support of the optional clusters shown here.

Table 25 – Clusters common to all devices

<table>
<thead>
<tr>
<th>Server side</th>
<th>Client side</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Mandatory</strong></td>
<td></td>
</tr>
<tr>
<td>Basic</td>
<td>None</td>
</tr>
<tr>
<td>Identify</td>
<td>None</td>
</tr>
<tr>
<td>Secure Communications</td>
<td>Secure Communications</td>
</tr>
<tr>
<td>Registration</td>
<td>Registration</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Optional</strong></td>
<td></td>
</tr>
<tr>
<td>Clusters with reporting capability (See 7.1.1 for details)</td>
<td>Clusters with reporting capability (See 7.1.1 for details)</td>
</tr>
<tr>
<td>Power Configuration [R2]</td>
<td>None</td>
</tr>
<tr>
<td>Inter-PAN Communication</td>
<td>Inter-PAN Communication</td>
</tr>
<tr>
<td>Alarms</td>
<td>None</td>
</tr>
<tr>
<td>Commissioning Cluster</td>
<td>Commissioning Cluster</td>
</tr>
<tr>
<td>Manufacturer-specific (See for 7.1.2 details)</td>
<td>Manufacturer-specific (See 7.1.2 for details)</td>
</tr>
</tbody>
</table>

7.1.1 Optional support for clusters with reporting capability

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

7.1.2 Manufacturer-specific clusters

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

7.1.3 Cluster usage restrictions

None.
7.2 Feature and function description

Each device must support a certain set of features and functions. Table 26 below is used to specify the mandatory and optional features and functions of each device. This chapter contains a description of what must be supported if the feature or function is supported by the device. The mandatory or optional configuration for each device is described in the upcoming chapters:

Table 26 Example features and functions configuration for an AMI device.

<table>
<thead>
<tr>
<th>Device Type/Feature or function</th>
<th>Join (end devices and routers only)</th>
<th>Form Network (coordinator only)</th>
<th>Allow Others to Join Network (routers and coordinators only)</th>
<th>Restore to Factory Fresh Settings</th>
<th>Pair Devices – (End Device Bind Request)</th>
<th>Bind Manager – (End Device Bind Response - Coordinator only)</th>
<th>Enable Identify Mode</th>
<th>Allow AMI devices to join the Network (routers and coordinators only)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mandatory/Optional</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>O</td>
<td>O</td>
<td>M</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>Device Type/Feature or function</td>
<td>Group Nodes (send out a Add group If Identify)</td>
<td>Create Scene (Store Scene)</td>
<td>Service discovery (Match Descriptor Request)</td>
<td>ZDP Bind Response</td>
<td>ZDP Unbind Response</td>
<td>End Device Annce/device Annce</td>
<td>Service Discovery response (Match Descriptor Response)</td>
<td>High Security Supported (ZigBee PRO only)</td>
</tr>
<tr>
<td>Mandatory/Optional</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>M</td>
<td>M</td>
<td>N/A</td>
</tr>
</tbody>
</table>

8 Join (end devices and routers):
   As described in Section 5.13.3.

9 Form Network (coordinator):
   As described in Section 5.13.3.

10 Allow Others to Join Network (Router and Coordinator only):
   As described in Section 5.13.3.

11 Restore to Factory fresh settings:
   As described in Section 5.13.3.

12 Pair Devices (End Device Bind Request):
   The device shall provide a way for the user to issue an End Device Bind Request.
Enable Identify mode:
The device shall provide a way for the user to enable Identify for 60 seconds.

Group Nodes (Add Group If Identify):
The device may provide a way for the user to send an “Add Group if Identifying Request.”

Create Scene (Store Scene):
The device may provide a way for the user to send a “Store Scene Req.”

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

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

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

End Device Annce/Device Annce:
The device shall Send End Device Annce (ZigBee 2006)/ Send Device (ZigBee 2007) upon joining and re-joining a network.

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

Allow AMI devices to Join the Network:
The device shall allow other AMI devices to join the network.

High Security supported: No
7.3 AMI Devices

7.3.1 Energy Service Portal

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

Roles:

- Able to get consumption data from meters and “store and forward” consumption data for the “sleeping” meters.
- Provide time synchronization services.
- Store historical consumption for in-home and EMS devices.
- Pass along informational messages and inter-PAN pricing data.
- Pass along demand response/load control messages to other AMI devices.

7.3.1.1 Supported clusters

In addition to those specified in Table 25, the Energy Service Portal device shall support the clusters listed in Table 27.

Table 27 – Clusters Supported by the Energy Service Portal

<table>
<thead>
<tr>
<th>Server side</th>
<th>Client side</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple Metering</td>
<td>Simple Metering</td>
</tr>
<tr>
<td>Message</td>
<td></td>
</tr>
<tr>
<td>Price</td>
<td></td>
</tr>
<tr>
<td>Demand Response/Load Control</td>
<td></td>
</tr>
<tr>
<td>Time</td>
<td></td>
</tr>
</tbody>
</table>

| Optional            |                   |
| Complex Metering    | Complex Metering  |
|                     |                   |
|                     | Price             |

|                   |                   |
| Prepayment         | Prepayment        |
7.3.1.2 Supported Features and Functions

The Energy Service Portal device shall have the features and functions listed below.

Table 28 Features and Functions supported by the Energy Service Portal.

<table>
<thead>
<tr>
<th>Device Type/Feature or function</th>
<th>Join (end devices and routers only)</th>
<th>Form Network (coordinator only)</th>
<th>Allow Others to Join Network (routers and coordinators only)</th>
<th>Restore to Factory Fresh Settings</th>
<th>Pair Devices – (End Device Bind Request)</th>
<th>Bind Manager – (End Device Bind Response - Coordinator only)</th>
<th>Enable Identify Mode</th>
<th>Allow AMI devices to join the Network (routers and coordinators only)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mandatory/Optional</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>O</td>
<td>O</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>Device Type/Feature or function</td>
<td>Group Nodes (send out a Add group If Identify)</td>
<td>Create Scene (Store Scene)</td>
<td>Service discovery (Match Descriptor Request)</td>
<td>ZDP Bind Response</td>
<td>ZDP Unbind Response</td>
<td>End Device Annce/device annce</td>
<td>Service Discovery response (Match Descriptor Response)</td>
<td>High Security Supported (ZigBee PRO only)</td>
</tr>
<tr>
<td>Mandatory/Optional</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>M</td>
<td>M</td>
<td>N/A</td>
</tr>
</tbody>
</table>

7.3.2 Metering Device

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

7.3.2.1 Supported clusters

In addition to those specified in Table 25, the Metering Device shall support the clusters listed in Table 29.

Table 29 – Clusters Supported by the Metering Device

<table>
<thead>
<tr>
<th>Server side</th>
<th>Client side</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Mandatory</strong></td>
<td></td>
</tr>
<tr>
<td>Simple Metering</td>
<td></td>
</tr>
<tr>
<td><strong>Optional</strong></td>
<td></td>
</tr>
<tr>
<td>Complex Metering</td>
<td></td>
</tr>
<tr>
<td>Time</td>
<td></td>
</tr>
<tr>
<td>Prepayment</td>
<td></td>
</tr>
<tr>
<td>Price</td>
<td></td>
</tr>
<tr>
<td>Message</td>
<td></td>
</tr>
</tbody>
</table>
7.3.2.2 Supported Features and Functions

The Metering Device shall have the features and functions listed below.

Table 30 Features and Functions supported by the Metering Device.

<table>
<thead>
<tr>
<th>Device Type/Feature or function</th>
<th>Join (end devices and routers only)</th>
<th>Form Network (coordinator only)</th>
<th>Allow Others to Join Network (routers and coordinators only)</th>
<th>Restore to Factory Fresh Settings</th>
<th>Pair Devices – (End Device Bind Request)</th>
<th>Bind Manager – (End Device Bind Response - Coordinator only)</th>
<th>Enable Identify Mode</th>
<th>Allow AMI devices to join the Network (routers and coordinators only)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mandatory/Optional</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>O</td>
<td>O</td>
<td>M</td>
<td>M</td>
<td>M</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Device Type/Feature or function</th>
<th>Group Nodes (send out a Add group If Identify)</th>
<th>Create Scene (Store Scene)</th>
<th>Service discovery (Match Descriptor Request)</th>
<th>ZDP Bind Response</th>
<th>ZDP Unbind Response</th>
<th>End Device Annce/device annce</th>
<th>Service Discovery response (Match Descriptor Response)</th>
<th>High Security Supported</th>
<th>(ZigBee PRO only)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mandatory/Optional</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>M</td>
<td>M</td>
<td>N/A</td>
<td></td>
</tr>
</tbody>
</table>

7.3.3 In-Premise Display

The In-Premise Display device will relay energy consumption data to the user by way of a graphical or text display. The display may or may not be an interactive device. At a minimum at least one of the following: current energy usage, a history over selectable periods, pricing information, or text messages. As an interactive device, it can be used for returning simple messages for interpretation by the recipient (e.g., “Button A was pressed”). The display may also show critical pricing information to advise the customer when peaks are due to occur so that they can take appropriate action.

7.3.3.1 Supported clusters

In addition to those specified in Table 25, the In-Premise Display device shall support the clusters listed in Table 31.

Table 31 – Clusters Supported by the In-Premise Display Device

<table>
<thead>
<tr>
<th>Server side</th>
<th>Client side</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mandatory</td>
<td>Demand Response &amp; load control</td>
</tr>
</tbody>
</table>

| Optional                                | Time                                     |

Page 66 Copyright © 2007, The ZigBee Alliance. All rights reserved. This is an unaccepted ZigBee specification draft, subject to change.
The device should state that at least one of the optional client clusters (Price, Simple Metering, or Messaging) must be implemented.

### 7.3.3.2 Supported Features and Functions

The In-Premise Display device shall have the features and functions listed below.

#### Table 32 and Functions supported by the In-Premise Display device.

<table>
<thead>
<tr>
<th>Device Type/Feature or function</th>
<th>Join (end devices and routers only)</th>
<th>Form Network (coordinator only)</th>
<th>Allow Others to Join Network (routers and coordinators only)</th>
<th>Restore to Factory Fresh Settings</th>
<th>Pair Devices – (End Device Bind Request)</th>
<th>Bind Manager – (End Device Bind Response - Coordinator only)</th>
<th>Enable Identify Mode</th>
<th>Allow AMI devices to join the Network (routers and coordinators only)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mandatory/Optional</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>O</td>
<td>O</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>Device Type/Feature or function</td>
<td>Group Nodes (send out a Add group If Identify)</td>
<td>Create Scene (Store Scene)</td>
<td>Service discovery (Match Descriptor Request)</td>
<td>ZDP Bind Response</td>
<td>ZDP Unbind Response</td>
<td>End Device Annce/device Annece</td>
<td>Service Discovery response (Match Descriptor Response)</td>
<td>High Security Supported (ZigBee PRO only)</td>
</tr>
<tr>
<td>Mandatory/Optional</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>M</td>
<td>M</td>
<td>N/A</td>
</tr>
</tbody>
</table>

### 7.3.4 Programmable Communicating Thermostat (PCT)

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

#### 7.3.4.1 Supported clusters

In addition to those specified in Table 25, the PCT device shall support the clusters listed in Table 33.

#### Table 33 – Clusters Supported by the PCT

<table>
<thead>
<tr>
<th>Server side</th>
<th>Client side</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mandatory</td>
<td></td>
</tr>
</tbody>
</table>
7.3.4.2 Supported Features and Functions

The PCT device shall have the features and functions listed below.

Table 34 Features and Functions supported by the PCT

<table>
<thead>
<tr>
<th>Device Type/Feature or function</th>
<th>Join (end devices and routers only)</th>
<th>Form Network (coordinator only)</th>
<th>Allow Others to Join Network (routers and coordinators only)</th>
<th>Restore to Factory Fresh Settings</th>
<th>Pair Devices – (End Device Bind Request)</th>
<th>Bind Manager – (End Device Bind Response - Coordinator only)</th>
<th>Enable Identify Mode</th>
<th>Allow AMI devices to join the Network (routers and coordinators only)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mandatory/Optional</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>O</td>
<td>O</td>
<td>M</td>
<td>M</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Device Type/Feature or function</th>
<th>Group Nodes (send out a Add group If Identify)</th>
<th>Create Scene (Store Scene)</th>
<th>Service discovery (Match Descriptor Request)</th>
<th>ZDP Bind Response</th>
<th>ZDP Unbind Response</th>
<th>End Device Annce/device Annce</th>
<th>Service Discovery response (Match Descriptor Response)</th>
<th>High Security Supported (ZigBee PRO only)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mandatory/Optional</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

7.3.5 Load Control

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

7.3.5.1 Supported clusters

In addition to those specified in Table 25, the Load Control device shall support the clusters listed in Table 35.

Table 35 – Clusters Supported by the Load Control Device

| Server side | Client side |
7.3.5.2 Supported Features and Functions

The Load Control Device shall support the features and functions listed below.

<table>
<thead>
<tr>
<th>Device Type/Feature or function</th>
<th>Join (end devices and routers only)</th>
<th>Form Network (coordinator only)</th>
<th>Allow Others to Join Network (routers and coordinators only)</th>
<th>Restore to Factory Fresh Settings</th>
<th>Pair Devices – (End Device Bind Request)</th>
<th>Bind Manager – (End Device Bind Response - Coordinator only)</th>
<th>Enable Identify Mode</th>
<th>Allow AML devices to join the Network (routers and coordinators only)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mandatory/Optional</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>O</td>
<td>O</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>Device Type/Feature or function</td>
<td>Group Nodes (send out a Add group If Identify)</td>
<td>Create Scene (Store Scene)</td>
<td>Service discovery (Match Descriptor Request)</td>
<td>ZDP Bind Response</td>
<td>ZDP Unbind Response</td>
<td>End Device Annce/device Annce</td>
<td>Service Discovery response (Match Descriptor Response)</td>
<td>High Security Supported (ZigBee PRO only)</td>
</tr>
<tr>
<td>Mandatory/Optional</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>M</td>
<td>M</td>
<td>N/A</td>
</tr>
</tbody>
</table>

7.3.6 Range Extender

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

7.3.6.1 Supported clusters

The Range Extender device shall only support the mandatory common clusters listed in Table 25.

<table>
<thead>
<tr>
<th>Server side</th>
<th>Client side</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mandatory</td>
<td></td>
</tr>
</tbody>
</table>
7.3.6.2 Supported Features and Functions

The Load Control device shall have the features and functions listed below.

<table>
<thead>
<tr>
<th>Device Type/Feature or function</th>
<th>Join (end devices and routers only)</th>
<th>Form Network (coordinator only)</th>
<th>Allow Others to Join Network (routers and coordinators only)</th>
<th>Restore to Factory Fresh Settings</th>
<th>Pair Devices – (End Device Bind Request)</th>
<th>Bind Manager – (End Device Bind Response - Coordinator only)</th>
<th>Enable Identify Mode</th>
<th>Allow AMI devices to join the Network (routers and coordinators only)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mandatory/Optional</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>O</td>
<td>O</td>
<td>M</td>
</tr>
<tr>
<td>Device Type/Feature or function</td>
<td>Group Nodes (send out a Add group If Identify)</td>
<td>Create Scene (Store Scene)</td>
<td>Service discovery (Match Descriptor Request)</td>
<td>ZDP Bind Response</td>
<td>ZDP Unbind Response</td>
<td>End Device Annce/device Annce</td>
<td>Service Discovery response (Match Descriptor Response)</td>
<td>High Security Supported (ZigBee PRO only)</td>
</tr>
<tr>
<td>Mandatory/Optional</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>M</td>
<td>M</td>
<td>N/A</td>
</tr>
</tbody>
</table>

7.3.7 Smart Appliance

Smart Appliances on the ZigBee network can participate in Energy Management activities. Examples of these are when Utilities initiates a demand response or pricing event, or appliance to actively inform customers (and possibly automated processes) via in-home displays of when or how energy is being used. In the latter case, scenarios include:

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

7.3.8 Smart Appliance

Smart Appliance devices on the ZigBee network can participate in energy management activities. Examples of these activities are when utilities initiate demand response or pricing events, or the appliance actively informs customers (and possibly automated processes) via in-home displays of when or how energy is being used. In the latter case, scenarios include:

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

In addition to those specified in Table 25 the Smart Appliance device shall support the clusters listed in Table 39.

<table>
<thead>
<tr>
<th>Table 39 – Clusters Supported by the Smart Appliance Device</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Server side</strong></td>
</tr>
<tr>
<td>Mandatory</td>
</tr>
<tr>
<td><strong>Optional</strong></td>
</tr>
<tr>
<td>Demand Response and Load Control</td>
</tr>
<tr>
<td>Price</td>
</tr>
<tr>
<td>Time</td>
</tr>
<tr>
<td>Message</td>
</tr>
</tbody>
</table>

The device should state that at least one of the optional client clusters (Demand Response and Load Control, Price, Simple Metering, or Messaging) must be implemented.

### 7.3.8.2 Supported Features and Functions

The Smart Appliance device shall have the features and functions listed below.

<table>
<thead>
<tr>
<th>Table 40 Features and Functions supported by the Smart Appliance device</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Device Type/Feature or function</strong></td>
</tr>
<tr>
<td><strong>Mandatory/Optional</strong></td>
</tr>
<tr>
<td><strong>Device Type/Feature or function</strong></td>
</tr>
<tr>
<td><strong>Mandatory/Optional</strong></td>
</tr>
</tbody>
</table>
7.3.9 Prepayment Terminal

The Prepayment Terminal device will allow utility customers or other users (e.g. sub-metered tenants) to pay for consumption in discrete increments rather than establishing a traditional billing agreement. The Prepayment Terminal device will accept payment (e.g. credit card, code entry), display remaining balances, and alert the user of a balance approaching zero, and may perform some or all of the other functions described in In-Premise Display.

7.3.9.1 Supported clusters

In addition to those specified in Table 25, the Prepayment Terminal device shall support the clusters listed in Table 41.

Table 41 – Clusters Supported by the Prepayment Terminal Device

<table>
<thead>
<tr>
<th>Server side</th>
<th>Client side</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mandatory</td>
<td>Optional</td>
</tr>
<tr>
<td>Price</td>
<td>Demand Response and Load Control</td>
</tr>
<tr>
<td>Time</td>
<td>Simple Metering</td>
</tr>
<tr>
<td>Prepayment</td>
<td>Message</td>
</tr>
</tbody>
</table>

7.3.9.2 Supported Features and Functions

The Prepayment Terminal device shall have the features and functions listed below.

Table 42 Features and Functions supported by the Prepayment Terminal device.

<table>
<thead>
<tr>
<th>Device Type/Feature or function</th>
<th>Join (end devices and routers only)</th>
<th>Form Network (coordinator only)</th>
<th>Allow Others to Join Network (routers and coordinators only)</th>
<th>Restore to Factory Fresh Settings</th>
<th>Pair Devices – (End Device Bind Request)</th>
<th>Bind Manager – (End Device Bind Response - Coordinator only)</th>
<th>Enable Identify Mode</th>
<th>Allow AMI devices to join the Network (routers and coordinators only)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mandatory/Optional</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>O</td>
<td>O</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>Device Type/Feature or function</td>
<td>Group Nodes (send out a Add group if Identify)</td>
<td>Create Scene (Store Scene)</td>
<td>Service discovery (Match Descriptor Request)</td>
<td>ZDP Bind Response</td>
<td>ZDP Unbind Response</td>
<td>End Device Annce/device Annce</td>
<td>Service Discovery response (Match Descriptor Response)</td>
<td>High Security Supported (ZigBee PRO only)</td>
</tr>
<tr>
<td>Mandatory/Optional</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>M</td>
<td>M</td>
<td>N/A</td>
</tr>
<tr>
<td>-------------------</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>----</td>
<td>----</td>
<td>-----</td>
</tr>
</tbody>
</table>

A Annex A - Candidate ZCL Material for use with this Profile

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

A.1 New Data Types

This section defines new ZCL data types needed for interoperability with AMI-based ZigBee devices and functions.

A.2 Definition of new types

The following material in this subsection is proposed for inclusion in the Data Types section (section 8.2) of the Foundation document [R3].

A.2.1 New Time Data Type

Table 43 – Additional Time Cluster Data type

<table>
<thead>
<tr>
<th>Type class</th>
<th>Data type ID</th>
<th>Data type</th>
<th>Length of data (octets)</th>
<th>Invalid number</th>
<th>Analog / Discrete</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>0xe2</td>
<td>UTCTime</td>
<td>4</td>
<td>0xffffffff</td>
<td>A</td>
</tr>
<tr>
<td>0xe3 – 0xe7</td>
<td>Reserved</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

A.2.1.1 UTCTime

UTCTime is an unsigned 32 bit value representing the number of seconds since 0 hrs, 0 minutes, 0 seconds, on the 1st of January, 2000 UTC. It reflects and defines the data type used in the ZCL Time Server attribute labeled as Time.

A.2.2 New Unsigned Integer Data Type

Table 44 – Additional Time Cluster Data type

<table>
<thead>
<tr>
<th>Type class</th>
<th>Data type ID</th>
<th>Data type</th>
<th>Length of data (octets)</th>
<th>Invalid number</th>
<th>Analog / Discrete</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unsigned Integer</td>
<td>0x24</td>
<td>Unsigned 40 Bit Integer</td>
<td>5</td>
<td>0xffffffff</td>
<td>A</td>
</tr>
<tr>
<td>0x25 – 0x27</td>
<td>Reserved</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
A.2.2.1 Unsigned 40 Bit Integer

This type represents an unsigned integer with a decimal range of 0 to $2^{40}-1$. The values that represents an invalid value of this type is 0xffffffff.
B Annex B – Inter-PAN Transmission Mechanism

B.1 Scope and Purpose

This document defines a mechanism whereby ZigBee devices can perform limited, insecure, and possibly anonymous exchanges of information with devices in their local neighborhood without having to form or join the same ZigBee network. The mandate for this feature comes from the Advanced Metering Infrastructure (AMI) TRD [R9]. The particular data exchange required by the AMI Application Profile is the request for anonymous public energy pricing information.

The intended destination for the mechanism described here is not the ZigBee specification [R1], but the relevant application profile documents for applications that make use of the feature – in particular, the AMI Profile Specification.

The material used to create Annex B is derived from [R10].

B.2 General description

B.2.1 What Inter-PAN Transmission Does

A schematic view of the how inter-PAN transmission in a ZigBee context works is shown in Figure 20.
Inter-PAN data exchanges are handled by a special “stub” of the Application Support Sub-Layer, which is accessible through a special Service Access Point (SAP), the INTRP-SAP, parallel to the normal APSDE-SAP. The stub APS performs just enough processing to pass application data frames to the MAC for transmission and to pass inter-PAN application frames from the MAC to the application on receipt.

**B.3 Service Specification**

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

- INTRP-DATA.request
- INTRP-DATA.confirm
- INTRP-DATA.indication

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

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

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

The primitive interface is as follows:

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

Parameters of the primitive appear in Table 45.
### Table 45 - Parameters of the INTRP-DATA.request

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Valid Range</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SrcAddrMode</td>
<td>Integer</td>
<td>0x02, 0x03</td>
<td>The addressing mode for the source address used in this primitive. This parameter can take one of the following values: 0x02 = 16-bit NWK address 0x03 = 64-bit extended address</td>
</tr>
<tr>
<td>DstAddrMode</td>
<td>Integer</td>
<td>0x01 – 0x03</td>
<td>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</td>
</tr>
<tr>
<td>DstPANID</td>
<td>16-bit PAN Id</td>
<td>0x0000 – 0xffff</td>
<td>The 16-bit PAN identifier of the entity or entities to which the ASDU is being transferred or the broadcast PANId 0xffff.</td>
</tr>
<tr>
<td>DstAddress</td>
<td>16-bit or 64-bit address</td>
<td>As specified by the AddrMode parameter</td>
<td>The address of the entity or entities to which the ASDU is being transferred.</td>
</tr>
<tr>
<td>ProfileId</td>
<td>Integer</td>
<td>0x0000 – 0xffff</td>
<td>The identifier of the application profile for which this frame is intended.</td>
</tr>
<tr>
<td>ClusterId</td>
<td>Integer</td>
<td>0x0000 – 0xffff</td>
<td>The identifier of the cluster, within the profile specified by the ProfileId parameter, which defines the application semantics of the ASDU.</td>
</tr>
<tr>
<td>ASDULength</td>
<td>Integer</td>
<td>0x00 – (\text{aMaxMACFrameSize} - 9)</td>
<td>The number of octets in the ASDU to be transmitted.</td>
</tr>
<tr>
<td>ASDU</td>
<td>Set of octets</td>
<td>-</td>
<td>The set of octets forming the ASDU to be transmitted.</td>
</tr>
<tr>
<td>ASDUHandle</td>
<td>Integer</td>
<td>0x00 – 0xfff</td>
<td>An integer handle associated with the ASDU to be transmitted.</td>
</tr>
</tbody>
</table>

#### B.3.1.2 When Generated

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

#### B.3.1.3 Effect on Receipt
On receipt of the INTRP-DATA.request primitive by the stub APS, the stub APS will construct and transmit a frame containing the given ASDU and other parameters using the MCPS-DATA.request primitive of the MAC sub-layer, as described in sub-clause 0, and, once the corresponding MCPS-DATA.confirm primitive is received, Generate the INTRP-DATA.confirm primitive with a status value reflecting the status value returned by the MAC.

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

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

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

The primitive interface is as follows:

```c
INTRP-DATA.confirm {
    ASDUHandle
    Status
}
```

Parameters of the primitive appear in Table 46.

**Table 46 - Parameters of the INTRP-DATA.confirm**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Valid Range</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASDUHandle</td>
<td>Integer</td>
<td>0x00 – 0xff</td>
<td>An integer handle associated with the transmitted frame.</td>
</tr>
<tr>
<td>Status</td>
<td>Enumeration</td>
<td>Any Status value returned by the MAC</td>
<td>The status of the ASDU transmission corresponding to ASDUHandle as returned by the MAC.</td>
</tr>
</tbody>
</table>

**B.3.2.2 When Generated**

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

**B.3.2.3 Effect on Receipt**

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

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

The INTRP-DATA.indication primitive allows the stub APS to inform the next higher layer that it has received a frame that was transmitted via the stub APS on another device.
B.3.3.1 Semantics of the Service Primitive

The primitive interface is as follows:

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

Parameters of the primitive appear in Table 47.
**Table 47 - Parameters of the INTRP-DATA.indication**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Valid Range</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SrcAddrMode</td>
<td>Integer</td>
<td>0x02, 0x03</td>
<td>The addressing mode for the source address used in this primitive. This parameter can take one of the following values: 0x02 = 16-bit NWK address 0x03 = 64-bit extended address</td>
</tr>
<tr>
<td>SrcPANId</td>
<td>16-bit PAN Id</td>
<td>0x0000 – 0xffff</td>
<td>The 16-bit PAN identifier of the entity from which the ASDU is being transferred.</td>
</tr>
<tr>
<td>SrcAddress</td>
<td>16-bit or 64-bit address</td>
<td>As specified by the SrcAddrMode parameter</td>
<td>The device address of the entity from which the ASDU is being transferred.</td>
</tr>
<tr>
<td>DstAddrMode</td>
<td>Integer</td>
<td>0x01 – 0x03</td>
<td>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</td>
</tr>
<tr>
<td>DstPANID</td>
<td>16-bit PAN Id</td>
<td>0x0000 – 0xffff</td>
<td>The 16-bit PAN identifier of the entity or entities to which the ASDU is being transferred or the broadcast PAN ID 0xffff.</td>
</tr>
<tr>
<td>DstAddress</td>
<td>16-bit or 64-bit address</td>
<td>As specified by the DstAddrMode parameter</td>
<td>The address of the entity or entities to which the ASDU is being transferred.</td>
</tr>
<tr>
<td>ProfileId</td>
<td>Integer</td>
<td>0x0000 – 0xffff</td>
<td>The identifier of the application profile for which this frame is intended.</td>
</tr>
<tr>
<td>ClusterId</td>
<td>Integer</td>
<td>0x0000 – 0xffff</td>
<td>The identifier of the cluster, within the profile specified by the ProfileId parameter, which defines the application semantics of the ASDU.</td>
</tr>
<tr>
<td>ASDULength</td>
<td>Integer</td>
<td>0x00 – ( a_{\text{MaxMACFrameSize}} - 9 )</td>
<td>The number of octets in the ASDU to be transmitted.</td>
</tr>
<tr>
<td>ASDU</td>
<td>Set of octets</td>
<td>-</td>
<td>The set of octets forming the ASDU to be transmitted.</td>
</tr>
<tr>
<td>LinkQuality</td>
<td>Integer</td>
<td>0x00 – 0xff</td>
<td>The link quality observed during the reception of the ASDU.</td>
</tr>
</tbody>
</table>

**B.3.3.2 When Generated**

---

Copyright © 2007, The ZigBee Alliance. All rights reserved. This is an unaccepted ZigBee specification draft, subject to change.
This primitive is generated and passed to the application in the event of the receipt, by the stub APS, of a MCPS-DATA.indication primitive from the MAC sub-layer, containing a frame that was generated by the stub APS of a peer ZigBee device, and that was intended for the receiving device.

**B.3.3.3 Effect on Receipt**

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

**B.4 Frame Formats**

The birds-eye view of a normal ZigBee frame is as shown in Figure 21.

<table>
<thead>
<tr>
<th>802.15.4 MAC Header</th>
<th>ZigBee NWK Header</th>
<th>ZigBee APS Header</th>
<th>ZigBee Payload</th>
</tr>
</thead>
</table>

**Figure 21 - Normal ZigBee Frame**

Briefly, the frame contains the familiar headers controlling the operation of the MAC sub-layer, the NWK layer and the APS. Following these, there is a payload, formatted as specified in [R3]

Since most of the information contained in the NWK and APS headers is not relevant for inter-PAN transmission, the inter-PAN frame, shown in Figure 23, contains only a stub of the NWK header the APS header, which provide the information required by the stub APS shown in Figure 23 to do its job.

<table>
<thead>
<tr>
<th>802.15.4 MAC Header</th>
<th>Stub NWK Header</th>
<th>Stub APS Header</th>
<th>ZigBee Payload</th>
</tr>
</thead>
</table>

**Figure 22 - Inter-PAN ZigBee Frame**

The stub NWK header is two octets in length and contains only a partial frame control field.

**Figure 23 - Stub NWK Header Format**

The format of the frame control field of the stub NWK header is formatted as shown in Figure 24.

<table>
<thead>
<tr>
<th>Bits: 0-1</th>
<th>2-5</th>
<th>6-15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frame type</td>
<td>Protocol version</td>
<td>Remaining sub-fields == 0</td>
</tr>
</tbody>
</table>
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 [R7].
- The value protocol version sub-field shall reflect the protocol version of the ZigBee stack as described in [R7].

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

The format of the stub APS header is shown in Figure 25.

```
Octets: 1       0/2       2       2
      APS frame control Group address Cluster identifier Profile identifier
                  Addressing fields
```

The stub APS header contains only 4 fields totaling a maximum of 7 octets in length.

The APS frame control field shall be 1 octet in length and is identical in format to the frame control field of the general APDU frame in [R7]. (see Figure 26).

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

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

- The frame type sub-field shall have a value of 0b11, which is a reserved frame type with respect to the [R7].
- The delivery mode sub-field may have a value of 0b00, indicating unicast, 0b10, indicating broadcast or 0b11 indicating group addressing.
- If the security sub-field has a value of 1 then the payload will be secured as described in [R7]. Otherwise the payload will be unsecured.
- The ACK request sub-field shall have a value of 0, indicating no ACK request, when the DstAddress the destination address field of the MAC header has a value of 0xffff and 1 otherwise.
- The extended header present sub-field shall always have a value of 0, indicating no extended header.
The optional group address shall be present if and only if the delivery mode field has a value of 0x0b11. If present it shall contain the 16-bit identifier of the group to which the frame is addressed.

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

The profile identifier is two octets in length and specifies the ZigBee profile identifier for which the frame is intended and shall be used during the filtering of messages at each device that takes delivery of the frame.

### B.5 Frame Processing

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

#### B.5.1 Inter-PAN Transmission

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

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

- The value of the SrcAddrMode parameter of the MCPS-DATA.request shall correspond to the value of the SrcAddrMode parameter of the INTRP-DATA.request.
- The SrcPANId parameter shall be equal to the value of the macPANID attribute of the MAC PIB.
- If the SrcAddrMode parameter has a value of 0x02 then the SrcAddr parameter shall be equal to the value of the MAC sub-layer constant aExtendedAddress. Otherwise it shall be equal to the value of the macShortAddress attribute of the MAC PIB.
- If the DstAddrMode parameter of the INTRP-DATA.request primitive has a value of 0x01 then the DstAddrMode parameter of the MCPS-DATA.request shall have a value of 0x02. Otherwise, the DstAddrMode parameter of the MCPS-DATA.request shall reflect the value of the DstAddrMode parameter of the INTRP-DATA.request.
- The DstPANId parameter shall have the value given by the DstPANID parameter of the INTRP-DATA.request primitive.
- If the DstAddrMode parameter of the INTRP-DATA.request has a value of 0x01, indicating group addressing, then the value of the DstAddr parameter of the MCPS-DATA.request shall be the broadcast address 0xffff. Otherwise, value of the DstAddr parameter shall reflect the value of the DstAddress parameter of the INTRP-DATA.request primitive.
- The MsduLength parameter shall be the length, in octets, of the stub APS frame.
- The Msdu parameter shall be the stub APS frame itself.
• If the transmission is a unicast then the value of the TxOptions parameter shall be 0x01, indicating a request for acknowledgement. Otherwise, the TxOptions parameter shall have a value of 0x00, indicating no options.

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

**B.5.2 Inter-PAN Reception**

On receipt of the MCPS-DATA.indication primitive from the MAC sub-layer, the receiving entity – in case of a ZigBee device this is normally the NWK layer – shall determine whether the frame should be passed to the stub APS or processed as specified in [R7]. For a frame that is to be processed by the stub APS, the non-varying sub-fields of both the NWK frame control field and the APS frame control field must be set exactly as described above.

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

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

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

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

• Similarly, the value of the SrcAddress parameter of the INTRP-DATA.indication shall reflect the value of the SrcAddr parameter of the MCPS-DATA.indication.

• Values for the DstAddrMode parameter shall be one of:
  - 0x03, if the DstAddrMode parameter of the MCPS-DATA.request has a value of 0x03.
  - 0x02, if the DstAddrMode parameter of the MCPS-DATA.request has a value of 0x02 and the delivery has a value other than 0x11.
  - 0x01, if the DstAddrMode parameter of the MCPS-DATA.request has a value of 0x02 and the delivery mode sub-field of the APS frame control field has a value of 0x11.

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

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

• The value of the ProfileId parameter shall be the same as the value of the profile identifier field of the stub APS header.
• The value of the ClusterId parameter shall be the same as the value of the cluster identifier field of the stub APS header.

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

• The ASDU shall be the stub APS payload itself.

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

B.6 Usage Scenario

Figure 27 shows a typical usage scenario for inter-PAN communication. In this AMI-oriented scenario, the Home Area Network (HAN) device is on the left with the APL, NWK and MAC shown as separate sequences. A ZigBee electric meter or Energy Service Portal (ESP) is also shown along with a “foreign”, i.e. non-ZigBee, device.

The first task of the HAN device in this scenario is to discover devices in the area that are capable of publishing pricing information. It could do this using an inter-PAN broadcast, i.e. a broadcast employing both the broadcast address and the broadcast PAN ID, but in doing this it runs the risk of confusing the non-ZigBee “foreign” device. As an alternative, the HAN device uses standard ZigBee network discovery (see [R7].) in order to find ZigBee PANs.

Once at least one ZigBee PAN is discovered, the HAN device sends a request for public pricing information using the INTRP-DATA SAP. Typically, the first time this request is sent, it will be sent as a broadcast to each discovered ZigBee PAN. Receiving devices that implement the INTRP-DATA SAP will process it and, if any such device is able to respond, it will respond directly to the requestor. After receiving at least one response the requestor may store the PAN ID and device address of one or more responders so that it may query them directly in future.
C Annex C – Key Establishment Cluster

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

C.1 Scope and Purpose

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

This Annex should be used in conjunction with the ZigBee Cluster Library, Foundation Specification (see [R3]), which gives an overview of the library and specifies the frame formats and general commands used therein.

This version is specifically for inclusion in the AMI profile. The document which originates from (075338r01) and (075375r00) will continue to be developed in a backward-compatible manner as a more general secure communication cluster for ZigBee applications as a whole.

C.2 General description

C.2.1 Introduction

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

C.2.2 Network security

The Key Establishment Cluster has been designed to be used where the underlying network security cannot be trusted. As such, no confidential information is transported in a manner which could be compromised.

To allow integrity and confidentiality of data passed between devices, cryptographic schemes need to be deployed.

C.2.3 Key Establishment

To allow integrity and confidentiality of data passed between devices, cryptographic schemes need to be deployed. The cryptographic scheme deployed in the ZigBee Specification for frame integrity and confidentiality is based upon a variant of the AES-CCM described in [R15] called AES-CCM*. This relies on the existence of secret keying material shared between the involved devices. There are methods to distribute this secret keying material in a trusted manner. However, these methods are generally not scalable or communication may be required with a trusted key allocation party over an insecure medium. This leads to the requirement for automated key establishment schemes to overcome these problems.

Key establishment schemes can either be effected using either a key agreement scheme or a key transport scheme. The key establishment scheme described in this document uses a key agreement scheme, therefore key transport schemes will not be considered further in this document.
A key agreement scheme is where both parties contribute to the shared secret and, therefore, the
derived secret keying material and the secret keying material to be established is not sent directly;
rather, information is exchanged between both parties that allows each party to derive the secret keying
material. Key agreement schemes may use either symmetric key or asymmetric key (public key)
techniques. The party that begins a key agreement scheme is called the initiator, and the other party is
called the responder.

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

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

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

- Symmetric Key Key Establishment
- Public Key Key Establishment

**C.2.3.1 Symmetric Key Key Establishment**

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

**C.2.3.2 Public Key Key Establishment**

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

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

CBKE provides the most comprehensive form of Key Establishment and therefore will be the method
specified in this cluster and is the scheme described in [R14].

The purpose of the key agreement scheme as described in this document is to produce shared secret
keying material which can subsequently used by devices using AES-CCM* the cryptographic scheme
deployed in the ZigBee Specification.

*Note RCC: Will keep it as general as possible so it should be possible to modify to accommodate all
potential schemes.
**C.2.3.3 General exchange**

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

![Diagram of general exchange](image)

**Figure 28 – Overview of general exchange**

The functions are as follows:

1. Exchange Ephemeral Data
2. Generate Key Bitstream
3. Derive SMAC key and Key Data
4. Confirm Key using SMAC

The functions shown in the diagram depend on the Key Establishment mechanism.

**C.2.3.3.1 Exchange Ephemeral Data**

The diagram shows ephemeral data $E_u$ and $E_v$. For SKKE schemes this is a random number and for PKKE schemes, the public key of a randomly generated public key pair.
ID_u and ID_v are the identifiers of the devices. For SKKE schemes this is usually the 64-bit device address [R11] and for PKKE schemes, a combination of the 64-bit device address [R11] and the device's static public key.

The ephemeral data E_u and the identifier ID_u is sent to V and the ephemeral data E_v and the identifier ID_v is sent to U.

**C.2.3.3.2 Generate Key Bitstream**

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

For SKKE schemes this is the Secret Key Generation (SKG) function based on the shared secret (master key) and for PKKE schemes, the ECMQV key generation function C(2,2, ECC MQV) with key confirmation scheme as specified in Sections 6.1.1 and 8.2 of SP 800-56A specification Error! Reference source not found.

**C.2.3.3.3 Derive SMAC key and key data**

The diagram shows the KeyDerive function for generating the SMAC Key and key data. The SMAC Key is used with a keyed hash message authentication function to generate a SMAC and the key data is the link key itself.

For SKKE schemes the key derivation function is as specified in Section 5.6.3 of [R13] ANSI X9.63-2001 and for PKKE schemes, the key derivation function is as specified in Section 6 of [R13] ANSI X9.63-2001.

**C.2.3.3.4 Confirm Key using SMAC**

The diagram shows SMACs MAC_u and MAC_v.

The SMAC MAC_u is sent to V and the SMAC MAC_v is sent to U. U and V both calculate the corresponding SMAC and compare it with the data sent.

**C.4 Cluster list**

The clusters specified in this document are listed in Table 48.

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

**Table 48 – Clusters specified for the Secure Communication functional domain**

<table>
<thead>
<tr>
<th>Cluster Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key establishment</td>
<td>Attributes and commands for establishing a shared secret between two ZigBee devices.</td>
</tr>
</tbody>
</table>

**Figure 29 – Typical Usage of the key establishment cluster**
C.2.1 Key establishment cluster

C.2.1.1 Overview

This cluster provides attributes and commands to perform mutual authentication and establish keys between two ZigBee devices.

C.2.1.2 Server

C.2.1.2.1 Dependencies

The Key Establishment server cluster has no dependencies.

C.2.1.2.2 Attributes

For convenience, the attributes defined in this specification are arranged into sets of related attributes; each set can contain up to 16 attributes. Attribute identifiers are encoded such that the most significant three nibbles specify the attribute set and the least significant nibble specifies the attribute within the set. The currently defined attribute sets are listed in Table 49.

Table 49 – Key establishment attribute sets

<table>
<thead>
<tr>
<th>Attribute set identifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x000</td>
<td>Information</td>
</tr>
<tr>
<td>0x001</td>
<td>Settings</td>
</tr>
<tr>
<td>0x002 – 0xfff</td>
<td>Reserved</td>
</tr>
</tbody>
</table>
C.2.1.2.3 Information

The Information attribute set contains the attributes summarized in Table 50.

Table 50 – Attributes of the Basic Device Information attribute set

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Name</th>
<th>Type</th>
<th>Range</th>
<th>Access</th>
<th>Default</th>
<th>Mandatory / Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0000</td>
<td>SecuritySuite</td>
<td>8-bit Enumeration</td>
<td>0x00 – 0xff</td>
<td>Read only</td>
<td>0x00</td>
<td>M</td>
</tr>
</tbody>
</table>

C.2.1.2.3.1 SecuritySuite attribute

The SecuritySuite attribute is 8-bits in length and specifies the cryptography scheme used for key establishment. This attribute shall be set to one of the non-reserved values listed in Table 51.

Table 51 – Values of the SecuritySuite attribute

<table>
<thead>
<tr>
<th>SecuritySuite</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>Certificate-based Key Establishment</td>
</tr>
</tbody>
</table>

C.2.1.2.4 Settings

TBC – configuration of cluster to be done here.

C.2.1.2.5 Commands Received

The server side of the key establishment cluster is capable of receiving the commands listed in Table 52.

Table 52 – Received Command IDs for the Key Establishment cluster

<table>
<thead>
<tr>
<th>Command identifier field value</th>
<th>Description</th>
<th>Mandatory / Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>Initiate Key Establishment</td>
<td>M</td>
</tr>
<tr>
<td>0x01</td>
<td>Confirm Key</td>
<td>M</td>
</tr>
<tr>
<td>0x02</td>
<td>Terminate Key Establishment</td>
<td>M</td>
</tr>
<tr>
<td>0x03 – 0xff</td>
<td>Reserved</td>
<td></td>
</tr>
</tbody>
</table>
C.2.1.2.5.1  Initiate Key Establishment command

The Initiate Key Establishment command allows the initiator sending device to initiate key establishment and transfer its identity (implicit certificate) and ephemeral data to the responder receiving device.

C.2.1.2.5.1.1  Payload Format

The Initiate command payload shall be formatted as illustrated in Figure 30 – Format of the Initiate Key Establishment command payload.

<table>
<thead>
<tr>
<th>Bits</th>
<th>384</th>
<th>176</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Type</td>
<td>Octet string</td>
<td>Octet string</td>
</tr>
<tr>
<td>Field Name</td>
<td>Implicit certificate (CERTU)</td>
<td>Ephemeral Data (QEU)</td>
</tr>
</tbody>
</table>

Figure 30 – Format of the Initiate Key Establishment command payload

Implicit Certificate field: The Implicit Certificate field shall be the octet-string representation of CERTU as specified in BC.3.2

Ephemeral Data field: The Ephemeral Data field shall be the octet-string representation of QEU as specified in section BC.3.2

C.2.1.2.5.1.2  Effect on Receipt

On receipt of this command, the responder device shall perform the implicit certificate check on the initiator sending device’s implicit certificate, and if successful send back its own implicit certificate and ephemeral data. It shall then generate an appropriate Initiate Key Establishment response command indicating success or failure. See C.2.1.2.6.1.

C.2.1.2.5.2  Confirm Key command

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

C.2.1.2.5.2.1  Payload Format

The Confirm Key command payload shall be formatted as illustrated in Figure 31

<table>
<thead>
<tr>
<th>Bits</th>
<th>128</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Type</td>
<td>Octet string</td>
</tr>
<tr>
<td>Field Name</td>
<td>Secure Message Authentication Code (SMACU)</td>
</tr>
</tbody>
</table>

Figure 31 – Format of the Confirm Key command payload
Secure Message Authentication Code field: The Secure Message Authentication Code field shall be the octet-string representation of SMACU as specified in section C.3.2.

C.2.1.2.5.2.2 Effect on Receipt

On receipt of this command, the responder device shall perform the implicit certificate check on the initiator sending device's implicit certificate, and if successful send back its own implicit certificate and ephemeral data. It shall then generate an appropriate Confirm Key response command indicating success or failure. See C.2.1.2.6.1.

C.2.1.2.5.3 Terminate Key Establishment command

The Terminate Key Establishment command allows the initiator sending device to terminate key establishment with the responder receiving device due to errors encountered at the initiator.

C.2.1.2.5.3.1 Payload Format

The Terminate Key Establishment command payload shall be formatted as illustrated Figure 32.

<table>
<thead>
<tr>
<th>Bits</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Type</td>
<td>8-bit Enumeration</td>
</tr>
<tr>
<td>Field Name</td>
<td>Status</td>
</tr>
</tbody>
</table>

Figure 32 – Format of the Terminate Key Establishment command payload

Status field: The Status field shall be one of the enumerated values in Table 53 – Status field.

<table>
<thead>
<tr>
<th>Enumeration</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUCCESS</td>
<td>0x00</td>
<td>The key establishment was successful.</td>
</tr>
<tr>
<td>BAD CERTIFICATE</td>
<td>0x01</td>
<td>The device's implicit certificate could not be verified.</td>
</tr>
<tr>
<td>BAD KEY CONFIRM</td>
<td>0x02</td>
<td>The device could not confirm that it shares the same key with the corresponding device.</td>
</tr>
<tr>
<td>BAD MESSAGE</td>
<td>0x03</td>
<td>The device received a bad message from the corresponding device (e.g., message with bad data, an out of sequence message, or a message with a bad format) and terminated the protocol.</td>
</tr>
<tr>
<td>TIMEOUT</td>
<td>0x04</td>
<td>The device timed out waiting for the corresponding device.</td>
</tr>
<tr>
<td>0x05 – 0xff</td>
<td></td>
<td>Reserved</td>
</tr>
</tbody>
</table>

C.2.1.2.5.3.2 Effect on Receipt

On receipt of this command, the device shall terminate key establishment with the initiator sender and clear all data generated during the key establishment process up to the point of reception.
C.2.1.2.6 Commands Generated

The server side of the authentication cluster is capable of generating the commands listed in Table 54 – Generated Command IDs for the Key Establishment cluster.

Table 54 – Generated Command IDs for the Key Establishment cluster

<table>
<thead>
<tr>
<th>Command identifier field value</th>
<th>Description</th>
<th>Mandatory / Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>Initiate Key Establishment response</td>
<td>M</td>
</tr>
<tr>
<td>0x01</td>
<td>Confirm Key response</td>
<td>M</td>
</tr>
<tr>
<td>0x02 – 0xff</td>
<td>Reserved</td>
<td></td>
</tr>
</tbody>
</table>

C.2.1.2.6.1 Initiate Key Establishment response command

The Initiate Key Establishment response command allows the sending device to respond to the initiator of key establishment and transfer its identity (implicit certificate) and ephemeral data to the receiving device.

C.2.1.2.6.1.1 Payload Format

The Initiate response command payload shall be formatted as illustrated in Figure 33 – Format of the Initiate Key Establishment response command payload. The Status field shall be set to ACCEPT or DECLINE.

<table>
<thead>
<tr>
<th>Bits</th>
<th>8</th>
<th>0/384</th>
<th>0/176</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Type</td>
<td>8-bit Enumeration</td>
<td>Octet string</td>
<td>Octet string</td>
</tr>
<tr>
<td>Field Name</td>
<td>Status</td>
<td>Implicit Certificate (CERTV)</td>
<td>Ephemeral Data (QEV)</td>
</tr>
</tbody>
</table>

Figure 33 – Format of the Initiate Key Establishment response command payload

Status field: The Status field shall be one of the enumerated values in Figure 33.

Implicit Certificate field: The Implicit Certificate field shall be the octet-string representation of CERTV as specified in section BC.3.2

Ephemeral Data field: The Ephemeral Data field shall be the octet-string representation of QEV as specified in section BC.3.2

C.2.1.2.6.2 Confirm Key response command

The Confirm Key response command allows the sending device to confirm the key established with the receiving device based on performing a cryptographic hash using part of the generated keying material and the identities and ephemeral data of both parties.
C.2.1.2.6.2.1 Payload Format

The Confirm Key command payload shall be formatted as illustrated in Figure 34.

<table>
<thead>
<tr>
<th>Bits</th>
<th>8</th>
<th>0/128</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Type</td>
<td>8-bit Enumeration</td>
<td>Octet string</td>
</tr>
<tr>
<td>Field Name</td>
<td>Status</td>
<td>Secure Message Authentication Code (SMACV)</td>
</tr>
</tbody>
</table>

Figure 34 – Format of the Confirm Key response command payload

Status field: The Status field shall be one of the enumerated values in Figure 34.

Secure Message Authentication Code field: The Secure Message Authentication Code field shall be the octet-string representation of SMACV as specified in section BC.3.2

C.2.1.3 Client

C.2.1.3.1 Dependencies

The Key Establishment server cluster has no dependencies.

C.2.1.3.2 Attributes

TBC.

C.2.1.3.3 Commands Received

The client receives the response commands detailed in sub-clause 0, as well as those used for reading and writing attributes.

C.2.1.3.4 Commands Generated

The client generates the commands detailed in sub-clause 0, as well as those used for reading and writing attributes.

C.3 Application implementation

C.3.1 Network security for AMI networks

The underlying network security for AMI networks is assumed to be ZigBee Standard security using pre-configured link keys.
A temporary link key for a joining device is produced by performing the cryptographic hash function on a random number assigned to the joining device (e.g. serial number) and the device identifier, which is the device's 64-bit IEEE address [R11].

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

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

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

**C.3.2 Certificate-Based Key Establishment**

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

The digital certificate includes:

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

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

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

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

- Both devices share a secret link key
- Implicit Key Authentication: Both devices know with whom they share this link key.
- Key Confirmation: Each device knows that the other device actually has computed the key correctly
- No Unilateral Key Control: No device has complete control over the shared link key that is established.
- Perfect Forward Secrecy: if the public key gets compromised none of future and past communications are exposed
- Known Key Security resilience: Each shared link key created per session is unique
C.3.2.1 Notation and representation

C.3.2.1.1 Strings and string operations

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

C.3.2.1.2 Integers and their representation

Throughout this specification, the representation of integers as bit strings or octet strings shall be fixed. All integers shall be represented as binary strings in most-significant-bit first order and as octet strings in most-significant-octet first order. This representation conforms to the convention in Section 4.3 of ANSI X9.63-2001 [R13].

C.3.2.1.3 Entities

Throughout this specification, each entity shall be a DEV and shall be uniquely identified by its 64-bit IEEE device address [R11]. The parameter \(entlen\) shall have the integer value 64.

C.3.2.2 Cryptographic building blocks

The following cryptographic primitives and data elements are defined for use with the CBKE protocol specified in this document.

C.3.2.2.1 Elliptic curve domain parameters

The elliptic curve domain parameters used in this specification shall be those for the curve “ansit163k1” as specified in Appendix J4.1, Example 1, of ANSI X9.63-2001 [R13]. All elliptic-curve points (and operations hereon) used in this specification shall be (performed) on this curve.

C.3.2.2.2 Elliptic-curve point representation

All elliptic-curve points shall be represented in polynomial notation as specified in Section 4.1.2.1 of ANSI X9.63-2001 [R13]. All elliptic-curve points shall be transmitted in compressed form, as specified in Section 4.2.2 of ANSI X9.63-2001 [R13]. Thus, each elliptic-curve point can be represented in 22 bytes.

C.3.2.2.3 Elliptic-curve public-key pair

An elliptic-curve-key pair consists of an integer \(q\) and a point \(Q\) on the curve determined by multiplying the generating point \(G\) of the curve by this integer (i.e., \(Q=qG\)) as specified in ANSI X9.63-2001 [R13]. Here, \(Q\) is called the public key, whereas \(q\) is called the private key; the pair \((q, Q)\) is called the public-key pair. Each private key shall be represented as specified in Section 4.3.1 of ANSI X9.63-2001 [R13]. Each public key shall be represented as defined in section <ref> of this document.
C.3.2.2.4 ECC implicit certificates

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

$$IC_U = \text{PublicReconstrKey} \ || \ \text{Subject} \ || \ \text{Issuer} \ || \ \text{ProfileAttributeData}$$

Where,

1. $\text{PublicReconstrKey}$: the 22-byte representation of the public-key reconstruction data $BEU$ as specified in the implicit certificate generation protocol, which is an elliptic-curve point as specified in section C.3.2.2.2;
2. $\text{Subject}$: the 8-byte identifier of the entity $U$ that is bound to the public-key reconstruction data $BEU$ during execution of the implicit certificate generation protocol (i.e., the extended, 64-bit IEEE 802.15.4 address [R11] of the device that purportedly owns the private key corresponding to the public key that can be reconstructed with $\text{PublicReconstrKey}$);
3. $\text{Issuer}$: the 8-byte identifier of the CA that creates the implicit certificate during the execution of the implicit certificate generation protocol (the so-called Certificate Authority);
4. $\text{ProfileAttributeData}$: the 10-byte sequence of octets that can be used by a ZigBee profile for any purpose. The first byte of this sequence is reserved as a profile identifier, which must be defined by another ZigBee standard.
5. The string $I_U$ as specified in Step 6 of the actions of the CA in the implicit certificate generation protocol (see section Error! Reference source not found.) shall be the concatenation of the $\text{Subject}$, $\text{Issuer}$, and $\text{ProfileAttributeData}$:

$$I_U = \text{Subject} \ || \ \text{Issuer} \ || \ \text{ProfileAttributeData}$$

Note RCC 2007-11-02: The final format of the certificate has not yet been decided as there are still decisions to be made on the CA hierarchy. However, if it is a case of a two-tier hierarchy, if there is a placeholder in the certificate for the identity of its CA then it should be possible to verify the binding of that CA to its public key using an additional certificate, which may be sufficient. The above format does contain such a placeholder.

C.3.2.2.5 ECC Block-cipher

The block-cipher used in this specification shall be the Advanced Encryption Standard AES-128, as specified in FIPS Pub 197 [R17]. This block-cipher has a key size that is equal to the block size: 128 bits.

C.3.2.2.6 ECC Cryptographic Hash Function

This is the Matyas-Meyer-Oseas hash function as specified in Annex B.6 in [R7].

C.3.2.2.7 ECC Keyed hash function for message authentication

The keyed hash message authentication code (HMAC) used in this specification shall be HMAC, as specified in the FIPS Pub 198 Error! Reference source not found., with the following instantiations:

1. Each entity shall use the cryptographic hash $H$ function as specified in section 0;
2. The block size $B$ shall have the integer value 16 (this block size specifies the length of the data integrity key, in bytes, that is used by the keyed hash function, i.e., it uses a 128-bit data integrity key);

The output size $\text{HMAClen}$ of the HMAC function shall have the same integer value as the message digest parameter $\text{hashlen}$ as specified in section.
C.3.2.3 CBKE key agreement scheme

The public-key key agreement protocols in this standard shall use the C(2,2, ECC MQV) with key confirmation scheme as specified in Sections 6.1.1 and 8.2 of SP 800-56A specification [R14], with the following instantiations:

1. Each entity shall be identified as specified in section 0;
2. Each entity shall use the HMAC-scheme as specified in section 0;
3. Each entity shall use the cryptographic hash function as specified in section 0;
4. The parameter keydatalen shall have the integer value 128;
5. The parameter SharedData shall be the empty string; parameter shareddatalen shall have the integer value 0;
6. The optional parameters Text1 and Text2 as specified as Text in Section 8.2 of SP 800-56A specification shall both be the empty string.
7. Each entity shall use the elliptic curve domain parameters as specified in section 0.

C.3.3 Certificate-Based Key Establishment

The CBKE method is used when the authenticity of both parties involved has not been established and where implicit authentication of both parties is required prior to key agreement.

The CBKE protocol has an identical structure to the PKKE protocol, except that implicit certificates are used rather than manual certificates. The implicit certificate protocol used with CBKE shall be the implicit certificate scheme with associated implicit certificate generation scheme and implicit certificate processing transformation as specified in SEC 4 [R20], with the following instantiations:

1. Each entity shall be a DEV;
2. Each entity’s identifier shall be its 64-bit device address [R11]; the parameter entlen shall have the integer value 64;
3. Each entity shall use the cryptographic hash function as specified in section 0;

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

1. Each entity shall have obtained information regarding the infrastructure that will be used for the operation of the implicit certificate scheme – including a certificate format and certificate generation and processing rules (see SEC 4 [R20]);
2. Each entity shall have access to an authentic copy of the elliptic-curve public keys of one or more certificate authorities that act as CA for the implicit certificate scheme (SEC 4 [R20]).

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

The methods used during the CBKE protocol are described below. The parameters used by these methods are described below.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Size (octets)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CERTU</td>
<td>48</td>
<td>The initiator device's implicit certificate used to transfer the public key (denoted (Q_0) in the in the SP 800-56A C(2,2, ECC MQV) specification [R14]).</td>
</tr>
</tbody>
</table>
### C.3.3.1 Exchange Ephemeral Data

#### C.3.3.1.1 Initiator

The initiator device's implicit certificate $CERTU$ and a newly generated ephemeral public key $QEU$ (see SP 800-56A Section 6.1.1 [R14]) are transferred to the responder device using the Initiate Key Establishment command via the Key Establishment Cluster Client.

#### C.3.3.1.2 Responder

The responder device's implicit certificate $CERTV$ and a newly generated ephemeral public key $QEV$ (see SP 800-56A Section 6.1.1 [R14]) are transferred to the initiator device using the Initiate Key Establishment response command via the Key Establishment Cluster Server.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Size (octets)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$CERTV$</td>
<td>48</td>
<td>The responder device's implicit certificate used to transfer the public key (denoted $Q_{e,v}$ in the SP 800-56A C(2,2, ECC MQV) specification [R14]).</td>
</tr>
<tr>
<td>$QEU$</td>
<td>22</td>
<td>The ephemeral public key generated by the initiator device (denoted $Q_{e,u}$ in the SP 800-56A C(2,2, ECC MQV) specification [R14]).</td>
</tr>
<tr>
<td>$QEV$</td>
<td>22</td>
<td>The ephemeral public key generated by the responder device (denoted $Q_{e,v}$ in the SP 800-56A C(2,2, ECC MQV) specification [R14]).</td>
</tr>
<tr>
<td>$MACU$</td>
<td>16</td>
<td>The secure message authentication code generated by the initiator device (denoted $SMAC(MacKey, MacLen, MacData_u)$ where $MacData_u = \text{message_string}_u</td>
</tr>
<tr>
<td>$MACV$</td>
<td>16</td>
<td>The secure message authentication code generated by the responder device (denoted $SMAC(MacKey, MacLen, MacData_v)$ where $MacData_v = \text{message_string}_v</td>
</tr>
</tbody>
</table>
C.3.3.2 Validate Implicit Certificates

C.3.3.2.1 Initiator

The initiator device’s Key Establishment Cluster Client processes the Initiate Key Establishment response command. The initiator device examines CERTV (formatted as IC_V as described in section BC.3.2.2.4), confirms that the Subject identifier is the purported owner of the certificate, and runs the certificate processing steps described in section Error! Reference source not found.

C.3.3.2.2 Responder

The responder device’s Key Establishment Cluster Server processes the Initiate Key Establishment command. The responder device examines CERTU (formatted as IC_U as described in section BC.3.2.2.4), confirms that the Subject identifier is the purported owner of the certificate, and runs the certificate processing steps described in section Error! Reference source not found.

C.3.3.3 Derive Keying Material

C.3.3.3.1 Initiator

The initiator device derives the keying material MacKey and KeyData as specified in see SP 800-56A Section 8.2 [R14].

C.3.3.3.2 Responder

The responder device derives the keying material MacKey and KeyData as specified in see SP 800-56A Section 8.2 [R14].

C.3.3.4 Confirm Keys

C.3.3.4.1 Initiator

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

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

C.3.3.4.2 Responder

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

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