Matter Device Library Specification

Version 1.4.2-adopted

Document:	23-27351-007_Matter-1.4.2-adopted-Device-Library-Specification.pdf
	July 16, 2025
Sponsored by:	Connectivity Standards Alliance
Accepted by:	This document has been accepted for release by the Connectivity Standards Alliance Board of Directors on **July 16, 2025**
Abstract:	The Matter Device Library Specification defines fundamental requirements for Matter Device Types.
Keywords:	Referenced in Chapter 1.

Matter Device Library

Version 1.4.2-adopted, 2025-07-24 10:34:49 -0700: Approved

Copyright Notice, License and Disclaimer

Elements of this document may be subject to third party intellectual property rights, including without limitation, patent, copyright or trademark rights, and any such third party may or may not be a member of the Connectivity Standards Alliance. Connectivity Standards Alliance members grant other Connectivity Standards Alliance members certain intellectual property rights as set forth in the Connectivity Standards Alliance IPR Policy. Connectivity Standards Alliance members do not grant you any rights under this license. The Connectivity Standards Alliance is not responsible for, and shall not be held responsible in any manner for, identifying or failing to identify any or all such third party intellectual property rights. Please visit www.csa-iot.org for more information on how to become a member of the Connectivity Standards Alliance.

This document and the information contained herein are provided on an "AS IS" basis and the Connectivity Standards Alliance DISCLAIMS ALL WARRANTIES EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO (A) ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OF THIRD PARTIES (INCLUDING WITHOUT LIMITATION ANY INTELLECTUAL PROPERTY RIGHTS INCLUDING PATENT, COPYRIGHT OR TRADEMARK RIGHTS); OR (B) ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE OR NONINFRINGEMENT. IN NO EVENT WILL THE CONNECTIVITY STANDARDS ALLIANCE BE LIABLE FOR ANY LOSS OF PROFITS, LOSS OF BUSINESS, LOSS OF USE OF DATA, INTERRUPTION OF BUSINESS, OR FOR ANY OTHER DIRECT, INDIRECT, SPECIAL OR EXEMPLARY, INCIDENTAL, PUNITIVE OR CONSEQUENTIAL DAMAGES OF ANY KIND, IN CONTRACT OR IN TORT, IN CONNECTION WITH THIS DOCUMENT OR THE INFORMATION CONTAINED HEREIN, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

All company, brand and product names in this document may be trademarks that are the sole property of their respective owners.

This notice and disclaimer must be included on all copies of this document.

Connectivity Standards Alliance
508 Second Street, Suite 206
Davis, CA 95616, USA

Revision History

Revision	Date	Details	Editor
1	September 23, 2022	Version 1.0	Robert Szewczyk
2	May 17, 2023	Version 1.1	Robert Szewczyk
3	October 18, 2023	Version 1.2	Robert Szewczyk
4	April 17, 2024	Version 1.3	Robert Szewczyk
5	November 4, 2024	Version 1.4	Robert Szewczyk
6	March 17, 2025	Version 1.4.1	Robert Szewczyk
7	July 16, 2025	Version 1.4.2	Robert Szewczyk

Table of Contents

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.

Connectivity Standards Alliance Reference Documents

Reference	Reference Location/URL	Description
[MatterCore]	https://groups.csa-iot.org/wg/members-all/document/27349	Matter Core Specification
[Alliance-PNP]	https://groups.csa-iot.org/wg/members/document/21624	Alliance Organizational Processes and Procedures, 13-0625
[Certification Policy]	https://groups.csa-iot.org/wg/members-all/document/125	Alliance Certification Policy, 15-0288

Provisional

Per [Alliance-PNP], when a specification is completed there may be sections of specification text (or smaller pieces of a section) that are not certifiable at this stage. These sections (or smaller pieces of a section) are marked as provisional prior to publishing the specification. This specification uses well-defined notation to mark Provisional Conformance (see [MatterCore], Section 7.3) or notes a section of text with the term "provisional".

List of Provisional Items

The following is a list of provisional items.

Chapter 1. Base Device Type

This chapter describes the base device type.

1.1. Base Device Type

1.1.1. Revision History

This is the revision history for this document. Because this document defines common requirements for all device types, changes to this document may affect many device types. Therefore, each device type definition affected by a change here, SHALL have its revision number incremented, with a new entry added to its history with a description that matches the description here.

Revision	Description
1	Initial revision
2	Duplicate condition replaces Multiple condition

1.1.2. Overview

This defines common conformance for all device types depending on, but not limited to:

- Certification programs (e.g. Zigbee, Matter, etc.)
- Underlying protocol stack (e.g. 802.15.4, Wi-Fi, Thread, Zigbee PRO, IPv6, TCP/IP)
- Regional regulations
- Interfaces (UI, cloud, etc.)
- Scale (e.g. residential vs commercial)
- Other common limitations or capabilities (e.g. battery powered or sleepy nodes).
- etc.

1.1.3. Conditions

Each section below is a category of conditions, each defining a list of conformance condition names and unique tags. The separation into categories is for reading purposes only.

1.1.3.1. Certification Program Conditions

At the time of the first publication of this document, many certification programs have terminated, or only allow re-certification, such as the Zigbee Home Automation standard.

Certification Program	Tag	Description
Zigbee Home Automation	ZHA	Zigbee Home Automation standard
Zigbee Smart Energy	ZSE	Zigbee Smart Energy standard

Certification Program	Tag	Description
Green Power	GP	Zigbee Green Power standard
Zigbee	Zigbee	Zigbee standard
SuZi	SuZi	Zigbee PRO Sub-GHz standard
Matter	Matter	Matter standard

1.1.3.2. Protocol Conditions

Protocol Tag
Ethernet
Wi-Fi
Thread
TCP
UDP
IP
IPv4
IPv6

1.1.3.3. Interface Conditions

Interface Tag	Description
LanguageLocale	The node supports localization for conveying text to the user
TimeLocale	The node supports localization for conveying time to the user
UnitLocale	The node supports localization for conveying units of measure to the user

Note that "supports localization" in the table above refers to supporting update of localization via cluster interactions.

1.1.4. Common Capability Conditions

This category is for common limitations or capabilities of a node.

Capability Tag	Description
SIT	The node is a short idle time intermittently connected device
LIT	The node is a long idle time intermittently connected device

Capability Tag	Description
Active	The node is always able to communicate

1.1.5. Device Type Class Conditions

This category is for classifications of device type. Some of these classifications are dependent on other conditions.

Class Tag	Summary
Node	the device type is classified as a Node device type (see Data Model specification)
App	the device type is classified as an Application device type (see Data Model specification)
Simple	the device type is classified as a Simple device type (see Data Model specification)
Dynamic	the device type is classified as a Dynamic device type (see Data Model specification)
Composed	the device type is composed of 2 or more device types (see System Model specification)

1.1.6. Endpoint Type Class Conditions

This category is for classifications of endpoints. Some of these classifications are dependent on other conditions.

Class Tag	Summary
Client	there exists a client application cluster on the endpoint
Server	there exists a server application cluster on the endpoint
Duplicate	the endpoint and at least one of its siblings have overlap in application device type(s)
BridgedPowerSourceInfo	the endpoint represents a Bridged Device, for which information about the state of its power source is available to the Bridge

1.1.6.1. Duplicate Condition

The endpoint and at least one of its sibling endpoints have an overlap in application device type(s), as defined in the "Disambiguation" section in the System Model specification. This condition triggers requirements for providing additional information about the endpoints in order to disambiguate between the endpoints (see "Disambiguation" section in the System Model specification).

1.1.7. Cluster Requirements

Each Matter device type implementation SHALL include these clusters, as a minimum set, based on the conformance defined below. This conformance table SHALL assume the Matter conformance condition is TRUE (in Conformance column).

ID	Cluster	Client/Server	Quality	Conformance
0x001D	Descriptor	Server		M
0x001E	Binding	Server		Simple & Client
0x0040	Fixed Label	Server		O
0x0041	User Label	Server		O

1.1.8. Element Requirements

Below list qualities and conformance that override the cluster specification requirements. A blank entry means no change.

ID	Cluster	Element	Name	Constraint	Access	Conformance
0x001D	Descriptor	Feature	TagList			Duplicate

Chapter 2. Utility Device Types

This chapter describes the utility device types. The utility device types are summarized in the table below:

Device ID	Device name
0x0016	Root Node
0x0011	Power Source
0x0012	OTA Requestor
0x0014	OTA Provider
0x0013	Bridged Node
0x0510	Electrical Sensor
0x050D	Device Energy Management
0x0019	Secondary Network Interface
0x0130	Joint Fabric Administrator

2.1. Root Node

This defines conformance for a root node endpoint (see System Model specification). This endpoint is akin to a "read me first" endpoint that describes itself and the other endpoints that make up the node.

- Device types with Endpoint scope SHALL NOT be supported on the same endpoint as this device type.
- Clusters with an Application role SHALL NOT be supported on the same endpoint as this device type.
- Other device types with Node scope MAY be supported on the same endpoint as this device type.

2.1.1. Revision History

This is the revision history for this device type. The highest revision number in the table below is the revision for this device type.

Revision	Description
1	Initial revision
2	Added Power Source to device type; Deprecated Power Source Configuration
3	Added restriction on Managed Device feature of Access Control cluster

2.1.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x0016	Root Node		Node	Node

2.1.3. Conditions

Condition	Description
CustomNetworkConfig	The node only supports out-of-band-configured networking (e.g. rich user interface, manufacturer-specific means, custom commissioning flows, or future IP-compliant network technology not yet directly supported by NetworkCommissioning cluster).
ManagedAclAllowed	The node has at least one endpoint where some Device Type present on the endpoint has a Device Library element requirement table entry that sets this condition to true.

See the Base Device Type definition for additional conformance tags.

2.1.4. Device Type Requirements

The table lists other device types to be implemented along with this device type based on conformance.

ID	Name	Constraint	Conformance
0x0011	Power Source		O

2.1.5. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

ID	Cluster	Client/Server	Quality	Conformance
0x0028	Basic Information	Server	I	M
0x001F	Access Control	Server	I	M
0x002E	Power Source Configuration	Server	I	O, D
0x0038	Time Synchronization	Server	I	O
0x003F	Group Key Management	Server	I	M

ID	Cluster	Client/Server	Quality	Conformance
0x0030	General Commissioning	Server	I	M
0x0031	Network Commissioning	Server		!CustomNetworkConfig
0x003C	Administrator Commissioning	Server	I	M
0x003E	Operational Credentials	Server	I	M
0x002B	Localization Configuration	Server	I	LanguageLocale
0x002C	Time Format Localization	Server	I	TimeLocale
0x002D	Unit Localization	Server	I	UnitLocale
0x0033	General Diagnostics	Server	I	M
0x0032	Diagnostic Logs	Server	I	O
0x0034	Software Diagnostics	Server	I	O
0x0037	Ethernet Network Diagnostics	Server		[Ethernet]
0x0036	Wi-Fi Network Diagnostics	Server		[Wi-Fi]
0x0035	Thread Network Diagnostics	Server		[Thread]
0x0046	ICD Management	Server	I	SIT | LIT

NOTE The Network Diagnostics clusters present on the Root Node SHALL serve the primary network interface as specified in the Network Commissioning cluster if it exists, or the out-of-band-configured networking interfaces.

2.1.6. Element Requirements

Below list qualities and conformance that override the cluster specification requirements. A blank table cell means there is no change to that item and the value from the cluster specification applies.

ID	Cluster	Element	Name	Constraint	Access	Conformance
0x001F	Access Control	Feature	MNGD	desc		[ManagedAclAllowed]

ID	Cluster	Element	Name	Constraint	Access	Conformance
0x0046	ICD Management	Feature	LongIdleTimeSupport			LIT

2.1.6.1. Access Control MNGD Conformance

The MNGD (Managed Device) feature of the Access Control Cluster on the device's Root Node endpoint is restricted to devices that contain an Application Endpoint type that explicitly permits its use, such as the Network Infrastructure Manager device type (Device Type ID 0x0090).

2.1.7. Endpoint Composition

A Root Node endpoint's Descriptor cluster PartsList attribute SHALL contain a list of all other endpoints on the node, i.e. the full-family pattern defined in the System Model specification.

2.2. Power Source

2.2.1. Classification

ID	Device Name	Superset Of	Class	Scope
0x0011	Power Source		Utility	Node

2.2.2. Revision History

Revision	Description
1	Initial revision

2.2.3. Cluster Requirements

This device SHALL support the clusters listed in the following table.

ID	ClusterName	Client/Server	Quality	Conformance
0x002F	Power Source	Server		M

2.3. OTA Requestor

An OTA Requestor is a device that is capable of receiving an OTA software update.

2.3.1. Revision History

This is the revision history for this device type. The highest revision number in the table below is the revision for this device type.

Revision	Description
1	Initial revision

2.3.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x0012	OTA Requestor		Utility	Node

2.3.3. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

ID	Cluster	Client/Server	Quality	Conformance
0x002A	OTA Software Update Requestor	Server		M
0x0029	OTA Software Update Provider	Client		M

2.4. OTA Provider

An OTA Provider is a node that is capable of providing an OTA software update to other nodes on the same fabric.

2.4.1. Revision History

This is the revision history for this device type. The highest revision number in the table below is the revision for this device type.

Revision	Description
1	Initial revision

2.4.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x0014	OTA Provider		Utility	Node

2.4.3. Cluster Requirements

Each node supporting this device type SHALL include these clusters based on the conformance defined below. A node SHALL only ever have, at most, one instance of the OTA Provider’s required clusters.

ID	Cluster	Client/Server	Quality	Conformance
0x002A	OTA Software Update Requestor	Client		O
0x0029	OTA Software Update Provider	Server		M

2.5. Bridged Node

This defines conformance for a Bridged Node root endpoint. This endpoint is akin to a "read me first" endpoint that describes itself and any other endpoints that make up the Bridged Node. A Bridged Node endpoint represents a device on a foreign network, but is not the root endpoint of the bridge itself.

2.5.1. Revision History

This is the revision history for this device type. The highest revision number in the table below is the revision for this device type.

Revision	Description
1	Initial revision
2	Added Power Source to device type; Deprecated Power Source Configuration
3	Added Ecosystem Information Cluster and FabricSynchronizedNode Condition.

2.5.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x0013	Bridged Node		Utility	Endpoint

2.5.3. Conditions

This device type MAY support the following conformance conditions as defined below.

Condition	Description
FabricSynchronizedNode	See description below.

See the Base Device Type definition for additional conformance tags.

2.5.3.1. FabricSynchronizedNode Condition

The FabricSynchronizedNode condition applies to a Bridged Node endpoint when all of the following are true:

- There is a Commissioner Control Cluster on an Aggregator which has this endpoint as a descendant.

- The Commissioner Control Cluster has a SupportedDeviceCategories attribute with the FabricSynchronization bit set.
- The bridged node is a Matter Node.

2.5.4. Device Type Requirements

This device type SHALL only be indicated on endpoints which are listed in the Descriptor cluster PartsList of another endpoint with an Aggregator device type.

The table lists other device types to be implemented along with this device type based on conformance.

ID	Name	Constraint	Conformance
0x0011	Power Source		O

2.5.5. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

ID	Cluster	Client/Server	Quality	Conformance
0x0039	Bridged Device Basic Information	Server		M
0x002E	Power Source Configuration	Server		BridgedPowerSourceInfo, D
0x002F	Power Source	Server		BridgedPowerSourceInfo
0x0750	Ecosystem Information	Server		FabricSynchronizedNode, O
0x003C	Administrator Commissioning	Server		FabricSynchronizedNode

2.5.6. Endpoint Composition

A Bridged Node endpoint SHALL support one of the following composition patterns:

- **Separate Endpoints**: All application device types are supported on separate descendant endpoints, and SHALL NOT be hosted on the Bridged Node endpoint. The Bridged Node endpoint’s Descriptor cluster PartsList attribute SHALL indicate a list of all endpoints representing the functionality of the bridged device, including the endpoints supporting the application device types, i.e. the full-family pattern defined in the System Model specification.
  This is used for the following cases:
  - Exposing a compound device - the child endpoints each have a part of the functionality of the bridged device. See endpoints 31-34 in the example below; the bridged device is a PIR

sensor which also has temperature and illuminance measurement. Endpoints 32-34 host the associated application device types and clusters. Endpoint 31 (the endpoint with the Bridged Node device type) functions as parent for these endpoints and has no application device types.

  - Exposing a composed device type - a child endpoint of the endpoint with the Bridged Node device type has the composed device type; this endpoint with the composed device type has child endpoints for the device type(s) that are mandatory or optional for the composed device type. See endpoints 41-43 in the example below; this is a refrigerator, which is a composed device type, hosted on endpoint 42, with the associated temperature controlled cabinet device type on child endpoint 43. Endpoint 41 (the endpoint with the Bridged Node device type) functions as parent for the endpoint hosting the composed device type and has no application clusters.
  - Combinations of the above.

- **One Endpoint**: Both the Bridged Node and one or more application device types are supported on the same endpoint (following application device type rules). The PartsList attribute in the Descriptor cluster SHALL be empty.
Since compound devices and composed device types each need more than one endpoint to expose their functionality, they cannot use the "One Endpoint" pattern and need to use the "Separate Endpoints" model described above.
  - Example in the figure below: endpoint 21 hosts the Bridged Node utility device type, plus the application device type for a dimmable light on same endpoint. Since the dimmable light device type is a superset of on/off light, that subset device type MAY be added here as well.

In all these composition patterns, endpoint composition SHALL conform to the application device type(s) definition.

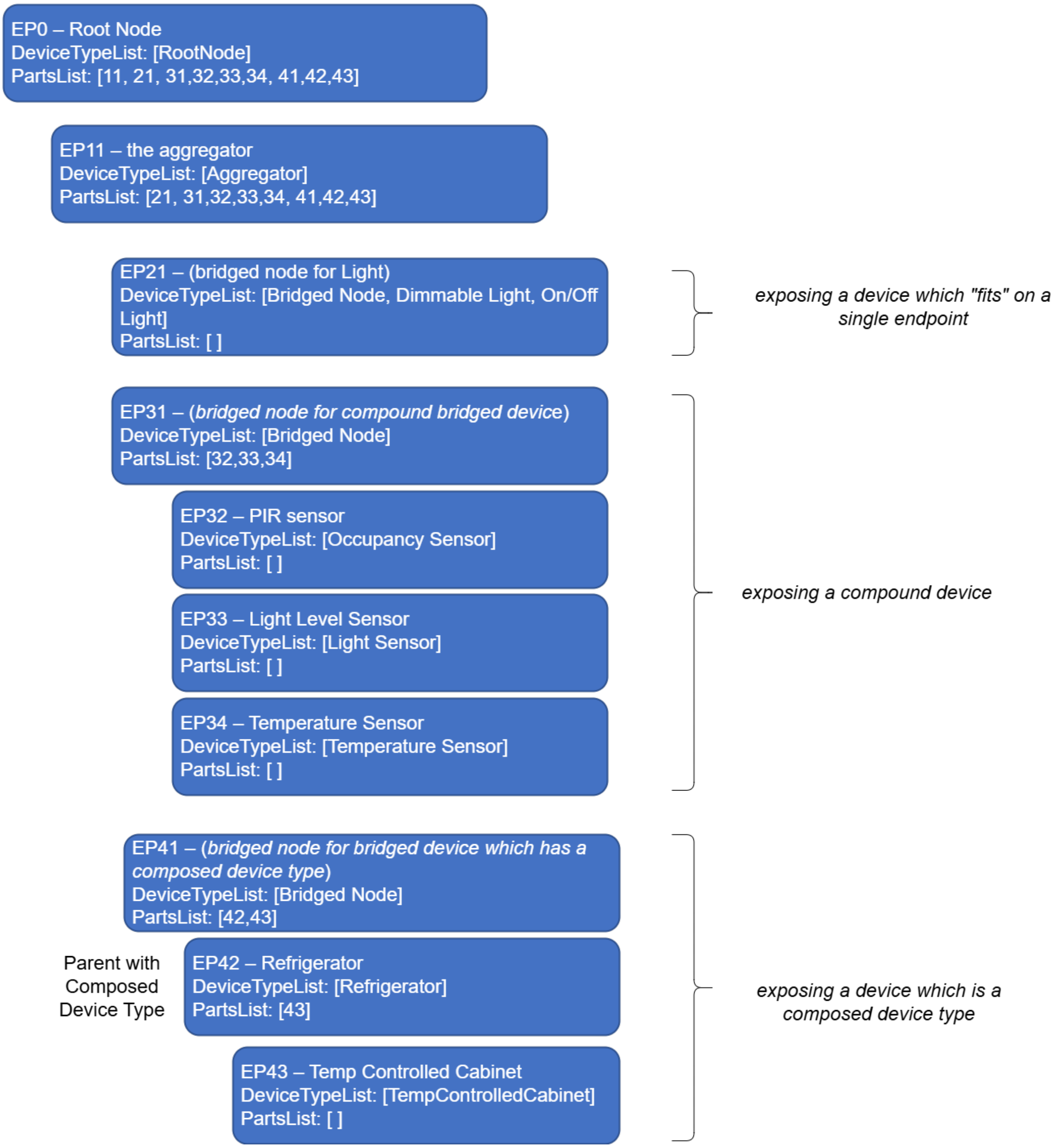

EP0 – Root Node
DeviceTypeList: [RootNode]
PartsList: [11, 21, 31,32,33,34, 41,42,43]
EP11 – the aggregator
DeviceTypeList: [Aggregator]
PartsList: [21, 31,32,33,34, 41,42,43]
EP21 – (bridged node for Light)
DeviceTypeList: [Bridged Node, Dimmable Light, On/Off Light]
PartsList: [ ]
exposing a device which "fits" on a single endpoint
EP31 – (bridged node for compound bridged device)
DeviceTypeList: [Bridged Node]
PartsList: [32,33,34]
EP32 – PIR sensor
DeviceTypeList: [Occupancy Sensor]
PartsList: [ ]
EP33 – Light Level Sensor
DeviceTypeList: [Light Sensor]
PartsList: [ ]
EP34 – Temperature Sensor
DeviceTypeList: [Temperature Sensor]
PartsList: [ ]
exposing a compound device
EP41 – (bridged node for bridged device which has a composed device type)
DeviceTypeList: [Bridged Node]
PartsList: [42,43]
Parent with Composed Device Type
EP42 – Refrigerator
DeviceTypeList: [Refrigerator]
PartsList: [43]
EP43 – Temp Controlled Cabinet
DeviceTypeList: [TempControlledCabinet]
PartsList: [ ]
exposing a device which is a composed device type

Figure 1. examples of composition for bridged nodes

2.6. Electrical Sensor

An Electrical Sensor device measures the electrical power and/or energy being imported and/or exported.

2.6.1. Revision History

Revision	Description
1	Initial revision

2.6.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x0510	Electrical Sensor		Utility	Endpoint

2.6.3. Device Type Requirements

Electrical measurements made by either the Electrical Power Measurement cluster, the Electrical Energy Measurement cluster, or both SHALL apply to the endpoints indicated by the Power Topology cluster.

2.6.4. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

ID	Name	Client/Server	Quality	Conformance
0x009C	Power Topology	Server		M
0x0090	Electrical Power Measurement	Server		O.a+
0x0091	Electrical Energy Measurement	Server		O.a+

2.7. Device Energy Management

A Device Energy Management device provides reporting and optionally adjustment of the electrical power planned on being consumed or produced by the device.

2.7.1. Revision History

Revision	Description
1	Initial revision
2	Updated description of when DEM Mode is to be included

2.7.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x050D	Device Energy Management		Utility	Endpoint

2.7.3. Conditions

See the Base Device Type definition for conformance tags.

Condition	Description
ControllableESA	The DEM cluster on this endpoint accepts commands to adjust its energy operation.

A ControllableESA device is one that allows a client to request either a change in power (PowerAdjustment feature), a change in the start time (StartTimeAdjustment feature), to be paused and resumed (Pausable feature), or to have its power or state forecast adjusted (ForecastAdjustment or ConstraintBasedAdjustment features).

Simple reporting of the Forecast as a single capability on its own (PowerForecastReporting or StateForecastReporting feature support) does not require the ControllableESA condition.

2.7.4. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

ID	Name	Client/Server	Quality	Conformance
0x0098	Device Energy Management	Server		M
0x009F	Device Energy Management Mode	Server		ControllableESA, O

2.7.5. Element Requirements

Below list qualities and conformance that override the cluster specification requirements. A blank table cell means there is no change to that item and the value from the cluster specification applies.

ID	Cluster	Element	Name	Constraint	Access	Conformance
0x0098	Device Energy Management	Feature	PowerAdjustment			[ControllableESA].a+
0x0098	Device Energy Management	Feature	StartTimeAdjustment			[ControllableESA].a+
0x0098	Device Energy Management	Feature	Pausable			[ControllableESA].a+
0x0098	Device Energy Management	Feature	ForecastAdjustment			[ControllableESA].a+

ID	Cluster	Element	Name	Constraint	Access	Conformance
0x0098	Device Energy Management	Feature	ConstraintBasedAdjustment			[ControllableESA].a+

2.8. Secondary Network Interface

A Secondary Network Interface device provides an additional network interface supported by the Node, supplementing the primary interface hosted by the Root Node endpoint.

A Node supporting multiple network interfaces SHALL include the primary interface on the Root Node endpoint, along with secondary interfaces on other endpoints. The priorities of these network interfaces are determined by the order of their endpoints, where interfaces with smaller endpoint numbers are higher priority.

2.8.1. Revision History

Revision	Description
1	Initial revision

2.8.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x0019	Secondary Network Interface		Utility	Endpoint

2.8.3. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

ID	Name	Client/Server	Quality	Conformance
0x0031	Network Commissioning	Server		M
0x0037	Ethernet Network Diagnostics	Server		[Ethernet]
0x0036	Wi-Fi Network Diagnostics	Server		[Wi-Fi]
0x0035	Thread Network Diagnostics	Server		[Thread]

NOTE The Network Diagnostics cluster present in this device type SHALL serve the secondary network interface as specified in the Network Commissioning cluster.

2.9. Joint Fabric Administrator

A Joint Fabric Administrator device provides capabilities to manage the Joint Fabric Datastore and issue an ICAC signed by the Joint Fabric Anchor Root CA.

A client wanting to access the capabilities of the Joint Fabric Administrator MAY use the Joint Commissioning Method (as specified in the Matter core specification) to be commissioned onto the Joint Fabric. Once commissioned, a client MAY access the capabilities of the Joint Fabric Administrator.

2.9.1. Joint Fabric Architecture

See the Joint Fabric section of the Multiple Fabrics chapter in the Matter core specification for more information on the Joint Fabric architecture.

2.9.2. Revision History

This is the revision history for this document.

Revision	Description
1	Initial revision

2.9.3. Classification

ID	Device Name	Superset Of	Class	Scope
0x0130	Joint Fabric Administrator		Utility	Endpoint

2.9.4. Cluster Requirements

Each endpoint supporting the Joint Fabric Administrator device type SHALL include these clusters based on the conformance defined below.

ID	Name	Client/Server	Quality	Conformance
0x0752	Joint Fabric Datastore	Server		M
0x0753	Joint Fabric Administrator	Server		M

Chapter 3. Application Device Types

The following chapters list the application device types defined in this version of the Device Library. They are grouped per functional area in a chapter and are summarized in the table below:

Device ID	Device name
lighting	
0x0100	On/Off Light
0x0101	Dimmable Light
0x010C	Color Temperature Light
0x010D	Extended Color Light
smart plugs/outlets and other actuators	
0x010A	On/Off Plug-in Unit
0x010B	Dimmable Plug-In Unit
0x010F	Mounted On/Off Control
0x0110	Mounted Dimmable Load Control
0x0303	Pump
0x0042	Water Valve
switches and controls	
0x0103	On/Off Light Switch
0x0104	Dimmer Switch
0x0105	Color Dimmer Switch
0x0840	Control Bridge
0x0304	Pump Controller
0x000F	Generic Switch
sensors	
0x0015	Contact Sensor
0x0106	Light Sensor
0x0107	Occupancy Sensor
0x0302	Temperature Sensor
0x0305	Pressure Sensor
0x0306	Flow Sensor
0x0307	Humidity Sensor
0x0850	On/Off Sensor
0x0076	Smoke CO Alarm

Device ID	Device name
0x002C	Air Quality Sensor
0x0041	Water Freeze Detector
0x0043	Water Leak Detector
0x0044	Rain Sensor
Entry Control	
0x000A	Door Lock
0x000B	Door Lock Controller
0x0202	Window Covering
0x0203	Window Covering Controller
HVAC	
0x0301	Thermostat
0x002B	Fan
0x002D	Air Purifier
0x030A	Thermostat Controller
media	
0x0028	Basic Video Player
0x0023	Casting Video Player
0x0022	Speaker
0x0024	Content App
0x0029	Casting Video Client
0x002A	Video Remote Control
generic	
0x0027	Mode Select
0x000E	Aggregator
robotic devices	
0x0074	Robotic Vacuum Cleaner
appliances	
0x0070	Refrigerator
0x0071	Temperature Controlled Cabinet
0x0072	Room Air Conditioner
0x0073	Laundry Washer
0x0075	Dishwasher
0x0077	Cook Surface

Device ID	Device name
0x0078	Cooktop
0x0079	Microwave Oven
0x007A	Extractor Hood
0x007B	Oven
0x007C	Laundry Dryer
energy	
0x050C	EVSE
0x050F	Water Heater
0x0017	Solar Power
0x0018	Battery Storage
0x0309	Heat Pump
network infrastructure	
0x0090	Network Infrastructure Manager
0x0091	Thread Border Router

Chapter 4. Lighting Device Types

4.1. On/Off Light

The On/Off Light is a lighting device that is capable of being switched on or off by means of a bound controller device such as an On/Off Light Switch or a Dimmer Switch. In addition, an on/off light is also capable of being switched by means of a bound occupancy sensor.

4.1.1. Revision History

Revision	Description
1	Initial Zigbee 3.0 revision
2	New data model format and notation
3	Updated the Scenes cluster to Scenes Management with Cluster ID: 0x0062

4.1.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x0100	On/Off Light		Simple	Endpoint

4.1.3. Conditions

See the Base Device Type definition for conformance tags.

4.1.4. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

Table 1. On/Off Light Cluster Requirements

ID	Cluster	Client/Server	Quality	Conformance
0x0003	Identify	Server		M
0x0004	Groups	Server		M
0x0062	Scenes Management	Server		M
0x0006	On/Off	Server		M
0x0008	Level Control	Server		O
0x0406	Occupancy Sensing	Client		O

The inclusion of the Level Control cluster on this device is recommended to provide a consistent user experience when the device is grouped with additional dimmable lights and the “with on/off” commands are used. For this device, since its only states are on or off, if the Level Control cluster is implemented, it SHALL NOT have any effect on the actual light level except for those commands that cause an on/off state change, that is, the “with on/off” commands. In addition, if the Level Control cluster is implemented, the device SHALL accept and process Level Control cluster commands, adjusting the value of the CurrentLevel attribute accordingly and, where necessary, adjusting the On/Off cluster OnOff attribute.

4.1.5. Element Requirements

Below list qualities and conformance that override the cluster specification requirements. A blank table cell means there is no change to that item and the value from the cluster specification applies.

ID	Cluster	Element	Name	Constraint	Access	Conformance
0x0003	Identify	Command	TriggerEffect			M
0x0062	Scenes Management	Command	CopyScene			M
0x0006	On/Off	Feature	Lighting			M
0x0008	Level Control	Feature	OnOff			M
0x0008	Level Control	Feature	Lighting			M
0x0008	Level Control	Attribute	CurrentLevel	1 to 254		
0x0008	Level Control	Attribute	MinLevel	1		
0x0008	Level Control	Attribute	MaxLevel	254		

As the TriggerEffect command of the Identify cluster and the OffWithEffect command of the On/Off cluster specify light effects that require dimming of the light output, and such is not possible on this device type, the specified light effects MAY be replaced by pure on/off light effects.

4.2. Dimmable Light

A Dimmable Light is a lighting device that is capable of being switched on or off and the intensity of its light adjusted by means of a bound controller device such as a Dimmer Switch or a Color Dimmer Switch. In addition, a Dimmable Light device is also capable of being switched by means of a bound occupancy sensor or other device(s).

4.2.1. Revision History

Revision	Description
1	Initial Zigbee 3.0 revision
2	New data model format and notation
3	Updated the Scenes cluster to Scenes Management with Cluster ID: 0x0062

4.2.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x0101	Dimmable Light	On/Off Light	Simple	Endpoint

4.2.3. Conditions

See the Base Device Type definition for conformance tags.

4.2.4. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

Table 2. Dimmable Light Cluster Requirements

ID	Cluster	Client/Server	Quality	Conformance
0x0003	Identify	Server		M
0x0004	Groups	Server		M
0x0062	Scenes Management	Server		M
0x0006	On/Off	Server		M
0x0008	Level Control	Server		M
0x0406	Occupancy Sensing	Client		O

4.2.5. Element Requirements

Below list qualities and conformance that override the cluster specification requirements. A blank table cell means there is no change to that item and the value from the cluster specification applies.

ID	Cluster	Element	Name	Constraint	Access	Conformance
0x0003	Identify	Command	TriggerEffect			M
0x0062	Scenes Management	Command	CopyScene			M
0x0006	On/Off	Feature	Lighting			M

ID	Cluster	Element	Name	Constraint	Access	Conformance
0x0008	Level Control	Feature	Lighting			M
0x0008	Level Control	Feature	OnOff			M
0x0008	Level Control	Attribute	CurrentLevel	1 to 254		
0x0008	Level Control	Attribute	MinLevel	1		
0x0008	Level Control	Attribute	MaxLevel	254		

4.3. Color Temperature Light

A Color Temperature Light is a lighting device that is capable of being switched on or off, the intensity of its light adjusted, and its color temperature adjusted by means of a bound controller device such as a Color Dimmer Switch.

4.3.1. Revision History

Revision	Description
1	Initial Zigbee 3.0 revision
2	New data model format and notation
3	Added optional occupancy sensing
4	Updated the Scenes cluster to Scenes Management with Cluster ID: 0x0062

4.3.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x010C	Color Temperature Light	Dimmable Light	Simple	Endpoint

4.3.3. Conditions

See the Base Device Type definition for conformance tags.

4.3.4. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

Table 3. Color Temperature Light Cluster Requirements

ID	Cluster	Client/Server	Quality	Conformance
0x0003	Identify	Server		M
0x0004	Groups	Server		M
0x0062	Scenes Management	Server		M
0x0006	On/Off	Server		M
0x0008	Level Control	Server		M
0x0300	Color Control	Server		M
0x0406	Occupancy Sensing	Client		O

4.3.5. Element Requirements

Below list qualities and conformance that override the cluster specification requirements. A blank table cell means there is no change to that item and the value from the cluster specification applies.

ID	Cluster	Element	Name	Constraint	Access	Conformance
0x0003	Identify	Command	TriggerEffect			M
0x0062	Scenes Management	Command	CopyScene			M
0x0006	On/Off	Feature	Lighting			M
0x0008	Level Control	Feature	OnOff			M
0x0008	Level Control	Feature	Lighting			M
0x0008	Level Control	Attribute	CurrentLevel	1 to 254		
0x0008	Level Control	Attribute	MinLevel	1		
0x0008	Level Control	Attribute	MaxLevel	254		
0x0300	Color Control	Feature	ColorTemperature			M
0x0300	Color Control	Attribute	RemainingTime			M

4.4. Extended Color Light

An Extended Color Light is a lighting device that is capable of being switched on or off, the intensity of its light adjusted, and its color adjusted by means of a bound controller device such as a Color Dimmer Switch or Control Bridge. The device supports adjustment of color by means of hue/saturation, enhanced hue, color looping, XY coordinates, and color temperature. In addition, the extended color light is also capable of being switched by means of a bound occupancy sensor.

4.4.1. Revision History

Revision	Description
1	Initial Zigbee 3.0 revision
2	New data model format and notation; integrate DM CCB 3501
3	Added optional occupancy sensing
4	Updated the Scenes cluster to Scenes Management with Cluster ID: 0x0062

4.4.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x010D	Extended Color Light	Color Temperature Light	Simple	Endpoint

4.4.3. Conditions

See the Base Device Type definition for conformance tags.

4.4.4. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

Table 4. Extended Color Light Cluster Requirements

ID	Cluster	Client/Server	Quality	Conformance
0x0003	Identify	Server		M
0x0004	Groups	Server		M
0x0062	Scenes Management	Server		M
0x0006	On/Off	Server		M
0x0008	Level Control	Server		M
0x0300	Color Control	Server		M

ID	Cluster	Client/Server	Quality	Conformance
0x0406	Occupancy Sensing	Client		O

4.4.5. Element Requirements

Below list qualities and conformance that override the cluster specification requirements. A blank table cell means there is no change to that item and the value from the cluster specification applies.

ID	Cluster	Element	Name	Constraint	Access	Conformance
0x0003	Identify	Command	TriggerEffect			M
0x0062	Scenes Management	Command	CopyScene			M
0x0006	On/Off	Feature	Lighting			M
0x0008	Level Control	Feature	OnOff			M
0x0008	Level Control	Feature	Lighting			M
0x0008	Level Control	Attribute	CurrentLevel	1 to 254		
0x0008	Level Control	Attribute	MinLevel	1		
0x0008	Level Control	Attribute	MaxLevel	254		
0x0300	Color Control	Feature	HueSaturation			O
0x0300	Color Control	Feature	EnhancedHue			O
0x0300	Color Control	Feature	ColorLoop			O
0x0300	Color Control	Feature	XY			M
0x0300	Color Control	Feature	ColorTemperature			M
0x0300	Color Control	Attribute	RemainingTime			M

Chapter 5. Smart Plugs/Outlets and other Actuators

5.1. On/Off Plug-in Unit

An On/Off Plug-in Unit is a device that provides power to another device that is plugged into it, and is capable of switching that provided power on or off.

The Mounted On/Off Control (added in Matter 1.4) has identical cluster requirements as the On/Off Plug-In Unit, and is marked as superset of this device type (since Matter 1.4.2). For devices intended to be mounted permanently, the Mounted On/Off Control device type SHALL be used, with the On/Off Plug-In Unit device type optionally added in the DeviceTypeList of the Descriptor cluster in addition to the On/Off Plug-In Unit device type (see [ref_MountedOnOffControlServerGuidance]).

Before Matter 1.4, mounted units typically used the On/Off Plug-In Unit device type. Clients can encounter devices which were made before or after these specification updates. Therefore, clients SHOULD use the following heuristic to distinguish the type of physical device based on the device type revision found on an endpoint ("--" means the device type is not listed).

On/Off Plug-In Unit (device type revision)	Mounted On/Off Control (device type revision)	Device Type
3 or lower	—	On/Off Plug-in Unit or Mounted On/Off Control (could be both)
4 or higher	—	On/Off Plug-in Unit
4 or higher	2 or higher	Mounted On/Off Control
—	2 or higher	Mounted On/Off Control (manufacturer not interested in older clients)
—	1	Mounted On/Off Control (Matter 1.4 did not have superset marking)
3	1	Mounted On/Off Control (CCB 4128 for a Matter 1.4/1.4.1 device)

5.1.1. Revision History

Revision	Description
1	Initial Zigbee 3.0 revision
2	New data model format and notation
3	Updated the Scenes cluster to Scenes Management with Cluster ID: 0x0062

Revision	Description
4	Add Mounted On/Off Control as superset, add usage guidance for both device types

5.1.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x010A	On/Off Plug-in Unit		Simple	Endpoint

5.1.3. Conditions

See the Base Device Type definition for conformance tags.

5.1.4. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

Table 5. On/Off Plug-in Unit Cluster Requirements

ID	Cluster	Client/Server	Quality	Conformance
0x0003	Identify	Server		M
0x0004	Groups	Server		M
0x0062	Scenes Management	Server		M
0x0006	On/Off	Server		M
0x0008	Level Control	Server		O
0x0406	Occupancy Sensing	Client		O

The inclusion of the Level Control cluster on this device is recommended to provide a consistent user experience when the device is grouped with additional dimmable lights and the “with on/off” commands are used. For this device, since its only states are on or off, if the Level Control cluster is implemented, it SHALL NOT have any effect on the actual light level except for those commands that cause an on/off state change, that is, the “with on/off” commands. In addition, if the Level Control cluster is implemented, the device SHALL accept and process Level Control cluster commands, adjusting the value of the CurrentLevel attribute accordingly and, where necessary, adjusting the On/Off cluster OnOff attribute.

5.1.5. Element Requirements

Below list qualities and conformance that override the cluster specification requirements. A blank table cell means there is no change to that item and the value from the cluster specification applies.

ID	Cluster	Element	Name	Constraint	Access	Conformance
0x0003	Identify	Command	TriggerEffect			M
0x0062	Scenes Management	Command	CopyScene			M
0x0006	On/Off	Feature	Lighting			M
0x0008	Level Control	Feature	OnOff			M
0x0008	Level Control	Feature	Lighting			M
0x0008	Level Control	Attribute	CurrentLevel	1 to 254		
0x0008	Level Control	Attribute	MinLevel	1		
0x0008	Level Control	Attribute	MaxLevel	254		

As the TriggerEffect command of the Identify cluster and the OffWithEffect command of the On/Off cluster specify light effects that require dimming of the light output, and such is not possible on this device type, the specified light effects MAY be replaced by pure on/off light effects.

5.2. Dimmable Plug-In Unit

A Dimmable Plug-In Unit is a device that provides power to another device that is plugged into it, and is capable of being switched on or off and have its level adjusted. The Dimmable Plug-in Unit is typically used to control a conventional non-communicating light through its mains connection using phase cutting.

The Mounted Dimmable Load Control (added in Matter 1.4) has identical cluster requirements as the Dimmable Plug-In Unit, and is marked as a superset of this device type (since Matter 1.4.2). For devices intended to be mounted permanently, the Mounted Dimmable Load Control device type SHALL be used, with the Dimmable Plug-In Unit device type optionally added to the DeviceTypeList of the Descriptor cluster in addition to the Mounted Dimmable Load Control device type (see [ref_MountedDimmableLoadControlServerGuidance]).

Before Matter 1.4, mounted dimmable load control units typically used the Dimmable Plug-In Unit device type. Clients can encounter devices which were made before or after these specification updates. Therefore, clients SHOULD use the following heuristic to distinguish the type of physical device based on the device type revision found on an endpoint ("--" means the device type is not listed).

Dimmable Plug-In Unit (device type revision)	Mounted Dimmable Load Control (device type revision)	Device Type
4 or lower	—	Dimmable Plug-in Unit or Mounted Dimmable Load Control (could be both)
5 or higher	—	Dimmable Plug-in Unit
5 or higher	2 or higher	Mounted Dimmable Load Control
—	2 or higher	Mounted Dimmable Load Control (manufacturer not interested in older clients)
—	1	Mounted Dimmable Load Control (Matter 1.4 did not have superset marking)
4	1	Mounted Dimmable Load Control (CCB 4128 for a Matter 1.4/1.4.1 device)

5.2.1. Revision History

Revision	Description
1	Initial Zigbee 3.0 revision
2	New data model format and notation
3	Added optional occupancy sensing
4	Updated the Scenes cluster to Scenes Management with Cluster ID: 0x0062
5	Add Mounted Dimmable Load Control as superset, add usage guidance for both device types

5.2.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x010B	Dimmable Plug-In Unit		Simple	Endpoint

5.2.3. Conditions

See the Base Device Type definition for conformance tags.

5.2.4. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

Table 6. Dimmable Plug-In Unit Cluster Requirements

ID	Cluster	Client/Server	Quality	Conformance
0x0003	Identify	Server		M
0x0004	Groups	Server		M
0x0062	Scenes Management	Server		M
0x0006	On/Off	Server		M
0x0008	Level Control	Server		M
0x0406	Occupancy Sensing	Client		O

5.2.5. Element Requirements

Below list qualities and conformance that override the cluster specification requirements. A blank table cell means there is no change to that item and the value from the cluster specification applies.

ID	Cluster	Element	Name	Constraint	Access	Conformance
0x0003	Identify	Command	TriggerEffect			M
0x0062	Scenes Management	Command	CopyScene			M
0x0006	On/Off	Feature	Lighting			M
0x0008	Level Control	Feature	OnOff			M
0x0008	Level Control	Feature	Lighting			M
0x0008	Level Control	Attribute	CurrentLevel	1 to 254		
0x0008	Level Control	Attribute	MinLevel	1		
0x0008	Level Control	Attribute	MaxLevel	254		

5.3. Mounted On/Off Control

A Mounted On/Off Control is a fixed device that provides power to another device or power circuit that is connected to it, and is capable of switching that provided power on or off.

This device type is intended for any wall-mounted or hardwired load controller, while On/Off Plug-in Unit is intended only for smart plugs and other power switching devices that are not permanently connected, and which can be unplugged from their power source.

> **NOTE**
>
> Since this device type was added in Matter 1.4, for endpoints using this device type it is RECOMMENDED to add the subset device type On/Off Plug-in Unit to the DeviceTypeList of the Descriptor cluster on the same endpoint for backward compatibility with existing clients.
> See [ref_MountedOnOffClientGuidance] for client guidance with these two device types.

5.3.1. Revision History

Revision	Description
1	Initial release
2	Add superset classification and usage guidance for both device types

5.3.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x010F	Mounted On/Off Control	On/Off Plug-in Unit	Simple	Endpoint

5.3.3. Conditions

See the Base Device Type definition for conformance tags.

5.3.4. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

Table 7. Mounted On/Off Control Cluster Requirements

ID	Cluster	Client/Server	Quality	Conformance
0x0003	Identify	Server		M
0x0004	Groups	Server		M
0x0062	Scenes Management	Server		M
0x0006	On/Off	Server		M
0x0008	Level Control	Server		O
0x0406	Occupancy Sensing	Client		O

The inclusion of the Level Control cluster on this device is recommended to provide a consistent user experience when the device is grouped with additional dimmable lights and the “with on/off” commands are used. For this device, since its only states are on or off, if the Level Control cluster is implemented, it SHALL NOT have any effect on the actual light level except for those commands that cause an on/off state change, that is, the “with on/off” commands. In addition, if the Level Control cluster is implemented, the device SHALL accept and process Level Control cluster commands, adjusting the value of the CurrentLevel attribute accordingly and, where necessary, adjusting the On/Off cluster OnOff attribute.

5.3.5. Element Requirements

Below list qualities and conformance that override the cluster specification requirements. A blank table cell means there is no change to that item and the value from the cluster specification applies.

ID	Cluster	Element	Name	Constraint	Access	Conformance
0x0003	Identify	Command	TriggerEffect			M
0x0062	Scenes Management	Command	CopyScene			M
0x0006	On/Off	Feature	Lighting			M
0x0008	Level Control	Feature	OnOff			M
0x0008	Level Control	Feature	Lighting			M
0x0008	Level Control	Attribute	CurrentLevel	1 to 254		
0x0008	Level Control	Attribute	MinLevel	1		
0x0008	Level Control	Attribute	MaxLevel	254		

As the TriggerEffect command of the Identify cluster and the OffWithEffect command of the On/Off cluster specify light effects that require dimming of the light output, and such is not possible on this device type, the specified light effects MAY be replaced by pure on/off light effects.

5.4. Mounted Dimmable Load Control

A Mounted Dimmable Load Control is a fixed device that provides power to a load connected to it, and is capable of being switched on or off and have its level adjusted. The Mounted Dimmable Load Control is typically used to control a conventional non-communicating light through its mains connection using phase cutting.

This device type is intended for any wall-mounted or hardwired dimmer-capable load controller, while Dimmable Plug-In Unit is intended only for dimmer-capable smart plugs that are not permanently connected, and which can be unplugged from their power source.

NOTE Since this device type was added in Matter 1.4, for endpoints using this device type

it is RECOMMENDED to add the subset device type Dimmable Plug-In Unit to the DeviceTypeList of the Descriptor cluster on the same endpoint for backward compatibility with existing clients.
See [ref_MountedDimmablePlugInUnitClientGuidance] for client guidance with these two device types.

5.4.1. Revision History

Revision	Description
1	Initial release
2	Add superset classification and usage guidance for both device types

5.4.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x0110	Mounted Dimmable Load Control	Dimmable Plug-In Unit	Simple	Endpoint

5.4.3. Conditions

See the Base Device Type definition for conformance tags.

5.4.4. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

Table 8. Mounted Dimmable Load Control Requirements

ID	Cluster	Client/Server	Quality	Conformance
0x0003	Identify	Server		M
0x0004	Groups	Server		M
0x0062	Scenes Management	Server		M
0x0006	On/Off	Server		M
0x0008	Level Control	Server		M
0x0406	Occupancy Sensing	Client		O

5.4.5. Element Requirements

Below list qualities and conformance that override the cluster specification requirements. A blank table cell means there is no change to that item and the value from the cluster specification applies.

ID	Cluster	Element	Name	Constraint	Access	Conformance
0x0003	Identify	Command	TriggerEffect			M
0x0062	Scenes Management	Command	CopyScene			M
0x0006	On/Off	Feature	Lighting			M
0x0008	Level Control	Feature	OnOff			M
0x0008	Level Control	Feature	Lighting			M
0x0008	Level Control	Attribute	CurrentLevel	1 to 254		
0x0008	Level Control	Attribute	MinLevel	1		
0x0008	Level Control	Attribute	MaxLevel	254		

5.5. Pump

A Pump device is a pump that may have variable speed. It may have optional built-in sensors and a regulation mechanism. It is typically used for pumping fluids like water.

5.5.1. Revision History

This is the revision history for this device type. The highest revision number in the table below is the revision for this device type.

Revision	Description
1	Initial Zigbee 3.0 revision
2	New data model format and notation
3	Updated the Scenes cluster to Scenes Management with Cluster ID: 0x0062

5.5.2. Classification

ID	Device name	Superset Of	Class	Scope
0x0303	Pump		Simple	Endpoint

5.5.3. Conditions

See the Base Device Type definition for conformance tags.

5.5.4. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

Table 9. Pump Cluster Requirements

ID	Cluster	Client/Server	Quality	Conformance
0x0006	On/Off	Server		M
0x0200	Pump Configuration and Control	Server		M
0x0003	Identify	Server		M
0x0008	Level Control	Server		O
0x0004	Groups	Server		O
0x0062	Scenes Management	Server		O
0x0402	Temperature Measurement	Server		O
0x0403	Pressure Measurement	Server		O
0x0404	Flow Measurement	Server		O
0x0402	Temperature Measurement	Client		O
0x0403	Pressure Measurement	Client		O
0x0404	Flow Measurement	Client		O
0x0406	Occupancy Sensing	Client		O

5.5.5. Cluster Restrictions

5.5.5.1. On/Off Cluster (Server) Clarifications

The actions carried out by a Pump device on receipt of commands are shown in the following.

Table 10. Pump Actions on Receipt for On/Off Commands

Command	Action on Receipt
Off	If the pump is powered on, store the current level then immediately power it off.

Command	Action on Receipt
On	If the pump is powered off, power it on and move immediately to the level stored by a previous Off command. If no such level has been stored, move immediately to the maximum level allowed for the pump.
Toggle	If the pump is powered on, proceed as for the Off command. If the device is powered off, proceed as for the On command.

5.5.5.2. Level Control Cluster (Server) Clarifications

The Level Control cluster SHALL allow controlling the pump setpoints. However, the transition time is always ignored.

The setpoint of the pump is a percentage related to the level according to the following table.

Table 11. Relationship between Level and Setpoint

Level	Setpoint	Meaning
0	N/A	Pump is stopped.
1–200	Level / 2 (0.5–100.0%)	Pump setpoint in percent.
201–255	100.0%	Pump setpoint is 100.0%

5.6. Water Valve Device Type

This defines conformance to the Water Valve device type.

5.6.1. Revision History

This is the revision history for this device type. The highest revision number in the table below is the revision for this device type.

Revision	Description
1	Initial revision

5.6.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x0042	Water Valve		Simple	Endpoint

5.6.3. Conditions

See the Base Device Type definition for conformance tags.

5.6.4. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

ID	Name	Client/Server	Quality	Conformance
0x0003	Identify	Server		M
0x0081	Valve Configuration and Control	Server		M
0x0404	Flow Measurement	Server		O
0x0404	Flow Measurement	Client		O

5.6.4.1. Identify Cluster

This cluster is used to identify the device.

5.6.4.2. Valve Configuration and Control Cluster

This cluster is used to configure and control (Open/Close) the valve.

5.6.4.3. Flow Measurement Cluster

The cluster server, if present, SHALL be used to report the measured flow through the valve.

The cluster client, if present, MAY be used via binding to close a control loop of flow through the valve.

5.6.5. Device implementation recommendations

5.6.5.1. Start Up Behavior

The start up behavior of a device with this device type, is currently not specified and is considered manufacturer specific. This means that the start up behavior and what is considered the "safe state", most suitable for the specific device, is defined by the manufacturer.

5.6.5.2. Firmware Update

When a device with this device type needs to update its firmware (or restart for another reason), it is strongly recommended to only perform the update/restart when the valve is in its closed state, as well as ignoring any open request during this update/restart, given the chance a valve can unintentionally be left in the open state, for longer periods of time.

Chapter 6. Switches and Controls Device Types

This Chapter specifies a number of "controller" device types like On/Off Light Switch and Dimmer Switch. Some products implementing these device types are intended to replace legacy switches or dimmers that directly control the power to a load. For such products, manufacturers are encouraged to implement an additional endpoint on the same product holding an "actuator" device type like an On/Off Light (or On/Off Plug-in Unit) or Dimmable Light (or Dimmable Plug-in Unit), consistent with the type of control it can provide to the load. In case product can control multiple loads separately, multiple such endpoints to each hold a device type for each load.

Additionally, having a central control function allows much richer automation triggered by a press of a switch. In such case, a switch works more like a sensor. For this, the Generic Switch device type is defined. See Section 6.6, “Generic Switch”. Manufacturers are encouraged to implement the Generic Switch device type as well in products that are generically referred to as switches. See Section 6.6.5, “Relation with other Switch device types (informative)” for examples how these device types can be combined.

6.1. On/Off Light Switch

An On/Off Light Switch is a controller device that, when bound to a lighting device such as an On/Off Light, is capable of being used to switch the device on or off.

6.1.1. Revision History

Revision	Description
1	Initial Zigbee 3.0 revision
2	New data model format and notation
3	Updated the Scenes cluster to Scenes Management with Cluster ID: 0x0062

6.1.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x0103	On/Off Light Switch		Simple	Endpoint

6.1.3. Conditions

See the Base Device Type definition for conformance tags.

6.1.4. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

Table 12. On/Off Light Switch Cluster Requirements

ID	Cluster	Client/Server	Quality	Conformance
0x0003	Identify	Server		M
0x0003	Identify	Client		M
0x0004	Groups	Client		O
0x0006	On/Off	Client		M
0x0062	Scenes Management	Client		O

6.2. Dimmer Switch

A Dimmer Switch is a controller device that, when bound to a lighting device such as a Dimmable Light, is capable of being used to switch the device on or off and adjust the intensity of the light being emitted.

6.2.1. Revision History

Revision	Description
1	Initial Zigbee 3.0 revision
2	New data model format and notation
3	Updated the Scenes cluster to Scenes Management with Cluster ID: 0x0062

6.2.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x0104	Dimmer Switch	On/Off Light Switch	Simple	Endpoint

6.2.3. Conditions

See the Base Device Type definition for conformance tags.

6.2.4. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

Table 13. Dimmer Switch Cluster Requirements

ID	Cluster	Client/Server	Quality	Conformance
0x0003	Identify	Server		M
0x0003	Identify	Client		M
0x0004	Groups	Client		O
0x0062	Scenes Management	Client		O
0x0006	On/Off	Client		M
0x0008	Level Control	Client		M

6.3. Color Dimmer Switch

A Color Dimmer Switch is a controller device that, when bound to a lighting device such as an Extended Color Light, is capable of being used to adjust the color of the light being emitted.

6.3.1. Revision History

Revision	Description
1	Initial Zigbee 3.0 revision
2	New data model format and notation
3	Updated the Scenes cluster to Scenes Management with Cluster ID: 0x0062

6.3.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x0105	Color Dimmer Switch	Dimmer Switch	Simple	Endpoint

6.3.3. Conditions

See the Base Device Type definition for conformance tags.

6.3.4. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

Table 14. Color Dimmer Switch Cluster Requirements

ID	Cluster	Client/Server	Quality	Conformance
0x0003	Identify	Server		M
0x0003	Identify	Client		M

ID	Cluster	Client/Server	Quality	Conformance
0x0004	Groups	Client		O
0x0062	Scenes Management	Client		O
0x0006	On/Off	Client		M
0x0008	Level Control	Client		M
0x0300	Color Control	Client		M

6.4. Control Bridge

A Control Bridge is a controller device that, when bound to a lighting device such as an Extended Color Light, is capable of being used to switch the device on or off, adjust the intensity of the light being emitted and adjust the color of the light being emitted. In addition, a Control Bridge device is capable of being used for setting scenes.

6.4.1. Revision History

Revision	Description
1	Initial Zigbee 3.0 revision
2	New data model format and notation
3	Updated the Scenes cluster to Scenes Management with Cluster ID: 0x0062

6.4.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x0840	Control Bridge		Simple	Endpoint

6.4.3. Conditions

See the Base Device Type definition for conformance tags.

6.4.4. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

Table 15. Control Bridge Cluster Requirements

ID	Cluster	Client/Server	Quality	Conformance
0x0003	Identify	Server		M
0x0003	Identify	Client		M
0x0004	Groups	Client		M

ID	Cluster	Client/Server	Quality	Conformance
0x0062	Scenes Management	Client		M
0x0006	On/Off	Client		M
0x0008	Level Control	Client		M
0x0300	Color Control	Client		M
0x0400	Illuminance Measurement	Client		O
0x0406	Occupancy Sensing	Client		O

6.5. Pump Controller

A Pump Controller device is capable of configuring and controlling a Pump device.

6.5.1. Revision History

Revision	Description
1	Initial Zigbee 3.0 revision
2	New data model format and notation
3	Updated the Scenes cluster to Scenes Management with Cluster ID: 0x0062
4	Remove Binding client as a result of review of CCB4058

6.5.2. Classification

ID	Device name	Superset Of	Class	Scope
0x0304	Pump Controller		Simple	Endpoint

6.5.3. Conditions

See the Base Device Type definition for conformance tags.

6.5.4. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

Table 16. Pump Controller Cluster Requirements

ID	Cluster	Client/Server	Quality	Conformance
0x0006	On/Off	Client		M

ID	Cluster	Client/Server	Quality	Conformance
0x0200	Pump Configuration and Control	Client		M
0x0003	Identify	Server		M
0x0003	Identify	Client		O
0x0004	Groups	Client		O
0x0008	Level Control	Client		O
0x0062	Scenes Management	Client		O
0x0402	Temperature Measurement	Client		O
0x0403	Pressure Measurement	Client		O
0x0404	Flow Measurement	Client		O

6.6. Generic Switch

This defines conformance for the Generic Switch device type.

6.6.1. Revision History

This is the revision history for this device type. The highest revision number in the table below is the revision for this device type.

Revision	Description
1	Initial revision
2	Removed requirement for Fixed Label cluster (instead use TagList which was added in Descriptor cluster)
3	Updated the Scenes cluster to Scenes Management with Cluster ID: 0x0062; note that this cluster is not used in this device type (only mentioned in the informative section related to On/Off Light Switch)

6.6.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x000F	Generic Switch		Simple	Endpoint

6.6.3. Conditions

See the Base Device Type definition for conformance tags.

6.6.4. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

ID	Cluster	Client/Server	Quality	Conformance
0x0003	Identify	Server		M
0x003B	Switch	Server		M

6.6.4.1. Instantaneous reporting

The generic mechanism for subscriptions and events might not ensure that detected interactions with the switch will be delivered "instantaneously" to the Switch client cluster in the interested party (they might be sent only after some time, e.g. due to batching of events and the `Min Interval` behavior for subscriptions). In order to achieve a good user experience, a device of this device type SHALL send updates of attributes and events defined in the Switch cluster without delay to subscribed parties.

6.6.4.2. Labeling for multi-switch devices

A Node which contains multiple switches will need to expose multiple endpoints each hosting an instance of this device type and the associated Switch cluster. This means the Duplicate condition in Matter base device requirements applies, so a TagList SHALL be included in the Descriptor cluster on each such endpoint. The tag(s) in this TagList are used to indicate orientation (e.g. left and right for a two-button switch) or labeling (e.g. "dim up" and "dim down" icons printed on the buttons) relevant to the user. A client SHOULD use these tags to convey such information to the user (e.g. showing it in a user interface), to help the user identify which endpoint maps to a certain orientation or labeling.

For the case where a server indicates tags from the Common Number Namespace, and the client presents entities related to the endpoints (e.g. icons for the various switches), it SHOULD present them in numerical order as indicated by the tags from the Common Number Namespace.

For a Node which has only one endpoint hosting an instance of this device type and the associated Switch cluster, a TagList MAY be used. This can be beneficial in cases where the switch has some user-recognizable labeling.

The TagList can contain a combination of tags from the namespaces defined in the Matter Semantic Tag Namespaces, including the namespace for switches as well as tags from a manufacturer-specific namespace.

In case the buttons have an intended function (e.g. engraved icon), the semantic tags from the Switches Namespace SHALL be used where applicable. If there is no corresponding tag, a manufacturer-specific tag with a string Label SHOULD be used (see Example 2 below).

To identify the location of a button on the device (e.g. top button of a two-button device), the semantic tags from the Common Position Namespace SHALL be used where applicable.

For devices where these are not applicable or not sufficient (e.g. a switch device with four buttons in a row), the semantic tags from the Common Number Namespace SHALL be used to enumerate the position of the buttons on the device, in left to right, top to bottom order, starting with `Number.One` for the first button.

For devices to control a Closure (e.g. Window Covering), the semantic tags from the Switches Namespace SHALL be used where applicable.

Example 1: a device with two rocker switches (mounted side by side), which has two endpoints (11,12) for the switch-related functionality

- endpoint 11 has device type Generic Switch and contains
  - cluster Switch (feature flags: LS) exposing the state and events of the left button
  - cluster Descriptor with its TagList containing two tags: `Position.Left` and `Number.One`
- endpoint 12 has device type Generic Switch and contains
  - cluster Switch (feature flags: LS) exposing the state and events of the right button
  - cluster Descriptor with its TagList containing two tags: `Position.Right` and `Number.Two`

If this device were to have labeling on the buttons like an "up" and "down" icon, the TagList would have a third tag (from the Switches Namespace) with values `Switches.Up` and `Switches.Down` respectively.

Example 2: a device with four push buttons (mounted in a square), each labeled with an icon for a certain scene setting, which has four endpoints (21,22,23,24) for the switch-related functionality

- endpoint 21 has device type Generic Switch and contains
  - cluster Switch (feature flags: MS) exposing the events of the top-left button
  - cluster Descriptor with its TagList containing four tags: `Position.Top`, `Position.Left`, `Number.One` and (Tag=`Switches.Custom`, Label="watch tv")
    - This last tag is a Switches.Custom tag accompanied with a label (the other three tags do not need a Label field).
- endpoint 22 has device type Generic Switch and contains
  - cluster Switch (feature flags: MS) exposing the events of the top-right button
  - cluster Descriptor with its TagList containing four tags: `Position.Top`, `Position.Right`, `Number.Two` and (Tag=`Switches.Custom`, Label="dinner")
- endpoint 23 has device type Generic Switch and contains
  - cluster Switch (feature flags: MS) exposing the events of the bottom-left button
  - cluster Descriptor with its TagList containing four tags: `Position.Bottom`, `Position.Left`, `Number.Three` and (Tag=`Switches.Custom`, Label="reading")
- endpoint 24 has device type Generic Switch and contains

    - cluster Switch (feature flags: MS) exposing the events of the bottom-right button
    - cluster Descriptor with its TagList containing four tags: `Position.Bottom`, `Position.Right`, `Number.Four` and (Tag=`Switches.Custom`, Label="nightlight")

6.6.5. Relation with other Switch device types (informative)

The Generic Switch device type and the On/Off Light Switch device type both convey information about interactions with a switch to another device.

- The On/Off Light Switch will send On/Off/Toggle commands from its On/Off (client) cluster to a device implementing the On/Off (server) cluster to control the on/off functionality of that device. An On/Off Light Switch device can also implement Groups and Scenes Management clusters and thus send group and scene commands. Basically, it is targeted at directly sending control commands to other devices. The binding table is used to tell the device where to send the commands.
- The Generic Switch device type will send updates of attributes (for Latching Switch only) and events to subscribed parties which implement the Switch client cluster, as indications of interaction with the switch - leaving the interpretation (e.g. which device should be actuated because of the interaction) to the subscribed party. So it can be compared to a sensor-type device.
  This allows a more comprehensive controller to combine the information from the switch with other inputs or information sources (e.g. time of day, user presence) to determine which control commands (e.g. on/off, scene recall, attribute change) are sent to other devices in the network.

A device manufacturer MAY implement both device types on the same switch device, to allow it to be used for both types of control, as in this example for a rocker switch which implements:

- endpoint 31 with device type On/Off Light Switch which contains
    - (client) cluster On/Off exposing the On/Off/Toggle commands
- endpoint 32 with device type Generic Switch which contains
    - (server) cluster Switch (feature flags: LS) exposing the state and events of the switch

When this device is used in a particular setup, binding tables and subscriptions can be used to determine how it is used:

- used as an On/Off Light Switch (no subscriptions to endpoint 32)
- used as a Generic Switch (no bindings on endpoint 31)
- used as both at the same time. In this case, an interaction with the switch would result in an On/Off/Toggle command being sent to devices listed in the binding table of endpoint 31, as well as attribute update and events being sent towards devices having a subscription with endpoint 32.

Chapter 7. Sensor Device Types

7.1. Contact Sensor Device Type

This defines conformance to the Contact Sensor device type.

7.1.1. Revision History

This is the revision history for this device type. The highest revision number in the table below is the revision for this device type.

Revision	Description
1	Initial revision
2	Add Boolean State Configuration as optional cluster

7.1.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x0015	Contact Sensor		Simple	Endpoint

7.1.3. Conditions

See the Base Device Type definition for conformance tags.

7.1.4. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

ID	Name	Client/Server	Quality	Conformance
0x0003	Identify	Server		M
0x0045	Boolean State	Server		M
0x0080	Boolean State Configuration	Server		O

7.1.4.1. Identify Cluster

This is used to identify the endpoint.

7.1.4.2. Boolean State Cluster

This is used to indicate the state of the sensor/detector.

The state of the Boolean State cluster SHALL reflect the sensor detection using this scheme:

Value	State
True	Closed or contact
False	Open or no contact

7.1.4.3. Boolean State Configuration Cluster

This is used to configure the sensor/detector and is for this device type linked to the configuration of the Boolean State cluster.

7.2. Light Sensor

A Light Sensor device is a measurement and sensing device that is capable of measuring and reporting the intensity of light (illuminance) to which the sensor is being subjected.

7.2.1. Revision History

Revision	Description
1	Initial Zigbee 3.0 revision
2	New data model format and notation
3	Restricting Groups client to Zigbee.

7.2.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x0106	Light Sensor		Simple	Endpoint

7.2.3. Conditions

See the Base Device Type definition for conformance tags.

7.2.4. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

Table 17. Light Sensor Cluster Requirements

ID	Cluster	Client/Server	Quality	Conformance
0x0003	Identify	Server		M
0x0004	Groups	Client		[Zigbee]
0x0400	Illuminance Measurement	Server		M

7.3. Occupancy Sensor Device Type

An Occupancy Sensor is a measurement and sensing device that is capable of measuring and reporting the occupancy state in a designated area.

7.3.1. Revision History

Revision	Description
1	Initial Zigbee 3.0 revision
2	New data model format and notation
3	Restricting Groups client to Zigbee
4	Add Boolean State Configuration as optional cluster

7.3.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x0107	Occupancy Sensor		Simple	Endpoint

7.3.3. Conditions

See the Base Device Type definition for conformance tags.

7.3.4. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

ID	Name	Client/Server	Quality	Conformance
0x0003	Identify	Server		M
0x0080	Boolean State Configuration	Server		O
0x0406	Occupancy Sensing	Server		M

7.3.4.1. Identify Cluster

This is used to identify the endpoint.

7.3.4.2. Boolean State Configuration Cluster

This is used to configure the sensor/detector (e.g. sensitivity) and is for this device type linked to the configuration of the Occupancy Sensing cluster on the same endpoint.

7.3.4.3. Occupancy Sensing Cluster

This is used to indicate occupancy as well as the type of occupancy sensor used for detection and configuring the delays related to the occupied and unoccupied transitions.

7.3.5. Multi-modality sensors

The Occupancy Sensing cluster defines multiple modalities that can be employed to sense occupancy. A device implementing multiple such modalities (exposed in the feature flags) can be implemented in two ways:

- A single endpoint with an Occupancy Sensing cluster which has two or more of these feature bits set to 1.
    - This requires reporting the combination the sensing results as a single bit in the Occupancy attribute (and the OccupancyChanged event, when supported), with a single set of timing parameters applied.
    - Sensitivity setting (via a Boolean State Configuration cluster on the same endpoint) applies to all the sensing modalities together via a manufacturer-specific mapping.
- Multiple endpoints each hosting an Occupancy Sensing cluster (each with one feature bit set):
    - The sensing result of each modality is reported separately in the Occupancy attribute (and the OccupancyChanged event, when supported) of each endpoint, governed by the set of timing parameters provided in the cluster on that endpoint.
        - This implies some of these attributes can have a different values than their counterparts on other endpoints and that a client MAY have to combine these values if it wants to derive a single value.
    - Each modality can be provided with an independent sensitivity setting via a Boolean State Configuration cluster located on one or more of the endpoints.

7.4. Temperature Sensor

A Temperature Sensor device reports measurements of temperature.

7.4.1. Revision History

This is the revision history for this device type. The highest revision number in the table below is the revision for this device type.

Revision	Description
1	Initial Zigbee 3.0 revision
2	New data model format and notation

7.4.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x0302	Temperature Sensor		Simple	Endpoint

7.4.3. Conditions

See the Base Device Type definition for conformance tags.

7.4.4. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

ID	Cluster	Client/Server	Quality	Conformance
0x0402	Temperature Measurement	Server		M
0x0003	Identify	Server		M
0x0004	Groups	Client		[Zigbee]

7.5. Pressure Sensor

A Pressure Sensor device measures and reports the pressure of a fluid.

7.5.1. Revision History

Revision	Description
1	Initial Zigbee 3.0 revision
2	New data model format and notation

7.5.2. Classification

ID	Device name	Superset Of	Class	Scope
0x0305	Pressure Sensor		Simple	Endpoint

7.5.3. Conditions

See the Base Device Type definition for conformance tags.

7.5.4. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

Table 18. Pressure Sensor Cluster Requirements

ID	Cluster	Client/Server	Quality	Conformance
0x0403	Pressure Measurement	Server		M
0x0003	Identify	Server		M
0x0004	Groups	Client		[Zigbee]

7.6. Flow Sensor

A Flow Sensor device measures and reports the flow rate of a fluid.

7.6.1. Revision History

Revision	Description
1	Initial Zigbee 3.0 revision
2	New data model format and notation

7.6.2. Classification

ID	Device name	Superset Of	Class	Scope
0x0306	Flow Sensor		Simple	Endpoint

7.6.3. Conditions

See the Base Device Type definition for conformance tags.

7.6.4. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

Table 19. Flow Sensor Cluster Requirements

ID	Cluster	Client/Server	Quality	Conformance
0x0404	Flow Measurement	Server		M
0x0003	Identify	Server		M
0x0004	Groups	Client		[Zigbee]

7.7. Humidity Sensor

A humidity sensor (in most cases a Relative humidity sensor) reports humidity measurements.

7.7.1. Revision History

This is the revision history for this device type. The highest revision number in the table below is the revision for this device type.

Revision	Description
1	Initial Zigbee 3.0 revision
2	New data model format and notation

7.7.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x0307	Humidity Sensor		Simple	Endpoint

7.7.3. Conditions

See the Base Device Type definition for conformance tags.

7.7.4. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

ID	Cluster	Client/Server	Quality	Conformance
0x0003	Identify	Server		M
0x0405	Relative Humidity Measurement	Server		M
0x0004	Groups	Client		[Zigbee]

7.8. On/Off Sensor

An On/Off Sensor is a measurement and sensing device that, when bound to a lighting device such as a Dimmable Light, is capable of being used to switch the device on or off.

7.8.1. Revision History

Revision	Description
1	Initial Zigbee 3.0 revision
2	New data model format and notation
3	Updated the Scenes cluster to Scenes Management with Cluster ID: 0x0062

7.8.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x0850	On/Off Sensor		Simple	Endpoint

7.8.3. Conditions

See the Base Device Type definition for conformance tags.

7.8.4. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

Table 20. On/Off Sensor Cluster Requirements

ID	Cluster	Client/Server	Quality	Conformance
0x0003	Identify	Server		M
0x0003	Identify	Client		M
0x0004	Groups	Client		O
0x0062	Scenes Management	Client		O
0x0006	On/Off	Client		M
0x0008	Level Control	Client		O
0x0300	Color Control	Client		O

7.9. Smoke CO Alarm

A Smoke CO Alarm device is capable of sensing smoke, carbon monoxide or both. It is capable of issuing a visual and audible alert to indicate elevated concentration of smoke or carbon monoxide.

Smoke CO Alarms are capable of monitoring themselves and issuing visual and audible alerts for hardware faults, critical low battery conditions, and end of service. Optionally, some of the audible alerts can be temporarily silenced. Smoke CO Alarms are capable of performing a self-test which performs a diagnostic of the primary sensor and issuing a cycle of the audible and visual life safety alarm indications.

Some smoke alarms MAY be capable of adjusting sensitivity. Smoke CO Alarm MAY have the ability to detect and report humidity levels, temperature levels, and contamination levels.

7.9.1. Revision History

This is the revision history for this device type. The highest revision number in the table below is the revision for this device type.

Revision	Description
1	Initial revision

7.9.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x0076	Smoke CO Alarm		Simple	Endpoint

7.9.3. Conditions

See the Base Device Type definition for conformance tags.

7.9.4. Device Type Requirements

A Smoke CO Alarm device type SHALL support an instance of a Power Source device type on some endpoint. See the Power Source cluster for more information.

ID	Name	Constraint	Conformance
0x0011	Power Source	min 1	M

7.9.5. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

ID	Cluster	Client/Server	Quality	Conformance
0x0003	Identify	Server		M
0x0004	Groups	Server		O
0x005C	Smoke CO Alarm	Server		M
0x0405	Relative Humidity Measurement	Server		O
0x0402	Temperature Measurement	Server		O
0x040C	Carbon Monoxide Concentration Measurement	Server		O

7.10. Air Quality Sensor

This defines conformance for the Air Quality Sensor device type.

An air quality sensor is a device designed to monitor and measure various parameters related to the quality of ambient air in indoor or outdoor environments.

7.10.1. Revision History

This is the revision history for this device type. The highest revision number in the table below is the revision for this device type.

Revision	Description
1	Initial revision

7.10.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x002C	Air Quality Sensor		Simple	Endpoint

7.10.3. Conditions

See the Base Device Type definition for conformance tags.

7.10.4. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

ID	Cluster	Client/Server	Quality	Conformance
0x0003	Identify	Server		M
0x005B	Air Quality	Server		M
0x0402	Temperature Measurement	Server		O
0x0405	Relative Humidity Measurement	Server		O
0x040C	Carbon Monoxide Concentration Measurement	Server		O
0x040D	Carbon Dioxide Concentration Measurement	Server		O
0x0413	Nitrogen Dioxide Concentration Measurement	Server		O
0x0415	Ozone Concentration Measurement	Server		O
0x042B	Formaldehyde Concentration Measurement	Server		O

ID	Cluster	Client/Server	Quality	Conformance
0x042C	PM1 Concentration Measurement	Server		O
0x042A	PM2.5 Concentration Measurement	Server		O
0x042D	PM10 Concentration Measurement	Server		O
0x042F	Radon Concentration Measurement	Server		O
0x042E	Total Volatile Organic Compounds Concentration Measurement	Server		O

7.11. Water Freeze Detector Device Type

This defines conformance to the Water Freeze Detector device type.

7.11.1. Revision History

This is the revision history for this device type. The highest revision number in the table below is the revision for this device type.

Revision	Description
1	Initial revision

7.11.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x0041	Water Freeze Detector		Simple	Endpoint

7.11.3. Conditions

See the Base Device Type definition for conformance tags.

7.11.4. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

ID	Name	Client/Server	Quality	Conformance
0x0003	Identify	Server		M

ID	Name	Client/Server	Quality	Conformance
0x0045	Boolean State	Server		M
0x0080	Boolean State Configuration	Server		O

7.11.4.1. Identify Cluster

This is used to identify the endpoint.

7.11.4.2. Boolean State Cluster

This is used to indicate the state of the sensor/detector.

The state of the Boolean State cluster SHALL reflect the sensor detection using this scheme of:

Value	State
True	Water could potentially freeze in the current ambient conditions
False	Water is very unlikely to freeze in the current ambient conditions

Due to the difficulty in quantifying the risk of freezing based on the dependency on external factors such as temperature, humidity, pressure, etc, the actual triggering of a detector of this type depends on the physical construction and characteristics of the device and is therefore considered manufacturer specific.

7.11.4.3. Boolean State Configuration Cluster

This is used to configure the sensor/detector and is for this device type linked to the configuration of the Boolean State cluster.

7.11.5. Element Requirements

The following table lists qualities and conformance that override the cluster specification requirements. A blank table cell means there is no change to that item and the value from the cluster specification applies.

ID	Cluster	Element	Name	Constraint	Access	Conformance
0x0045	Boolean State	Event	StateChange			M

7.12. Water Leak Detector Device Type

This defines conformance to the Water Leak Detector device type.

7.12.1. Revision History

This is the revision history for this device type. The highest revision number in the table below is the revision for this device type.

Revision	Description
1	Initial revision

7.12.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x0043	Water Leak Detector		Simple	Endpoint

7.12.3. Conditions

See the Base Device Type definition for conformance tags.

7.12.4. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

ID	Name	Client/Server	Quality	Conformance
0x0003	Identify	Server		M
0x0045	Boolean State	Server		M
0x0080	Boolean State Configuration	Server		O

7.12.4.1. Identify Cluster

This is used to identify the endpoint.

7.12.4.2. Boolean State Cluster

This is used to indicate the state of the sensor/detector.

The state of the Boolean State cluster SHALL reflect the sensor detection using this scheme of:

Value	State
True	Water leak detected
False	No water leak detected

7.12.4.3. Boolean State Configuration Cluster

This is used to configure the sensor/detector and is for this device type linked to the configuration of the Boolean State cluster.

7.12.5. Element Requirements

The following table lists qualities and conformance that override the cluster specification requirements. A blank table cell means there is no change to that item and the value from the cluster specification applies.

ID	Cluster	Element	Name	Constraint	Access	Conformance
0x0045	Boolean State	Event	StateChange			M

7.13. Rain Sensor Device Type

This defines conformance to the Rain Sensor device type.

7.13.1. Revision History

This is the revision history for this device type. The highest revision number in the table below is the revision for this device type.

Revision	Description
1	Initial revision

7.13.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x0044	Rain Sensor		Simple	Endpoint

7.13.3. Conditions

See the Base Device Type definition for conformance tags.

7.13.4. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

ID	Name	Client/Server	Quality	Conformance
0x0003	Identify	Server		M
0x0045	Boolean State	Server		M
0x0080	Boolean State Configuration	Server		O

7.13.4.1. Identify Cluster

This is used to identify the endpoint.

7.13.4.2. Boolean State Cluster

This is used to indicate the state of the sensor/detector.

The state of the Boolean State cluster SHALL reflect the sensor detection using this scheme of:

Value	State
True	Rain detected
False	No rain detected

7.13.4.3. Boolean State Configuration Cluster

This is used to configure the sensor/detector and is for this device type linked to the configuration of the Boolean State cluster.

7.13.5. Element Requirements

The following table lists qualities and conformance that override the cluster specification requirements. A blank table cell means there is no change to that item and the value from the cluster specification applies.

ID	Cluster	Element	Name	Constraint	Access	Conformance
0x0045	Boolean State	Event	StateChange			M

Chapter 8. Entry Control Device Types

Entry Control device types are device types that control entry to a space, e.g. by people, pets, sun, wind, or rain.

8.1. Door Lock Device Type

A Door Lock is a device used to secure a door. It is possible to actuate a door lock either by means of a manual or a remote method.

8.1.1. Revision History

Revision	Description
1	Initial Zigbee 3.0 revision
2	Initial Matter revision
3	Updated the Scenes cluster to Scenes Management with Cluster ID: 0x0062; as noted below, this cluster remains disallowed in this device type

8.1.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x000A	Door Lock		Simple	Endpoint

8.1.3. Conditions

See the Base Device Type definition for conformance tags.

8.1.4. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

ID	Name	Client/Server	Quality	Conformance
0x0003	Identify	Server		M
0x0004	Groups	Server		X
0x0062	Scenes Management	Server		X
0x0101	Door Lock	Server		M

Nodes supporting this device type MAY implement Matter Time Synchronization by including the Time Synchronization Cluster (0x0038) server on the Root Node Endpoint and MAY also implement a Time Synchronization Cluster client for querying time from other nodes. Nodes supporting this device type that implement a Time Synchronization Cluster client SHALL include the Time Synchronization Cluster server on the Root Node Endpoint and SHALL include the Time Synchronization Client Feature.

8.1.5. Element Requirements

Below list qualities and conformance that override the cluster specification requirements. A blank table cell means there is no change to that item and the value from the cluster specification applies.

ID	Cluster	Element	Name	Constraint	Access	Conformance
0x001F	Access Control	Attribute	Extension			M
0x0101	Door Lock	Feature	User			PIN | RID | FGP | FACE | ALIRO
0x0101	Door Lock	Feature	RFIDCredential			P, O

8.2. Door Lock Controller Device Type

A Door Lock Controller is a device capable of controlling a door lock.

8.2.1. Revision History

Revision	Description
1	Initial Zigbee 3.0 revision
2	Initial Matter revision
3	Updated the Scenes cluster to Scenes Management with Cluster ID: 0x0062

8.2.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x000B	Door Lock Controller		Simple	Endpoint

8.2.3. Conditions

See the Base Device Type definition for conformance tags.

8.2.4. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

ID	Name	Client/Server	Quality	Conformance
0x0004	Groups	Client		O
0x0062	Scenes Management	Client		O
0x0038	Time Synchronization	Server		O
0x0101	Door Lock	Client		M

8.3. Window Covering

This defines conformance to the Window Covering device type.

8.3.1. Revision History

This is the revision history for this device type. The highest revision number in the table below is the revision for this device type.

Revision	Description
1	Initial Zigbee 3.0 revision
2	New data model format and notation
3	Updated the Scenes cluster to Scenes Management with Cluster ID: 0x0062
4	Scenes Management cluster was removed from Cluster Requirements

8.3.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x0202	Window Covering		Simple	Endpoint

8.3.3. Conditions

See the Base Device Type definition for conformance tags.

8.3.4. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

ID	Cluster	Client/Server	Quality	Conformance
0x0003	Identify	Server		M
0x0004	Groups	Server		Active, O
0x0102	Window Covering	Server		M

8.3.5. Element Requirements

Below list qualities and conformance that override the cluster specification requirements. A blank table cell means there is no change to that item and the value from the cluster specification applies.

ID	Cluster	Element	Name	Constraint	Access	Conformance
0x0102	Window Covering	Feature	AbsolutePosition			Zigbee

8.4. Window Covering Controller

A Window Covering Controller is a device that controls an automatic window covering.

8.4.1. Revision History

Revision	Description
1	Initial Zigbee 3.0 revision
2	New data model format and notation
3	Updated the Scenes cluster to Scenes Management with Cluster ID: 0x0062
4	Scenes Management cluster was removed from Cluster Requirements

8.4.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x0203	Window Covering Controller		Simple	Endpoint

8.4.3. Conditions

See the Base Device Type definition for conformance tags.

8.4.4. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

ID	Cluster	Client/Server	Quality	Conformance
0x0003	Identify	Server		O
0x0003	Identify	Client		O
0x0004	Groups	Client		Active, O
0x0102	Window Covering	Client		M

8.4.5. Element Requirements

Below list qualities and conformance that override the cluster specification requirements. A blank table cell means there is no change to that item and the value from the cluster specification applies.

ID	Cluster	Element	Name	Constraint	Access	Conformance
0x0102	Window Covering	Feature	AbsolutePosition			Zigbee

Chapter 9. HVAC Device Types

9.1. Thermostat

A Thermostat device is capable of having either built-in or separate sensors for temperature, humidity or occupancy. It allows the desired temperature to be set either remotely or locally. The thermostat is capable of sending heating and/or cooling requirement notifications to a heating/cooling unit (for example, an indoor air handler) or is capable of including a mechanism to control a heating or cooling unit directly.

9.1.1. Revision History

This is the revision history for this device type. The highest revision number in the table below is the revision for this device type.

Revision	Description
1	Initial Zigbee 3.0 revision
2	New data model format and notation, added Clusters required for Matter support, restricted legacy elements to Zigbee only
3	Addition of Energy Preference cluster and updated the Scenes cluster to Scenes Management with Cluster ID: 0x0062
4	Remove Time Synchronization cluster, Scenes Management and Zigbee only clusters, remove provisional marking from Energy Preference cluster

9.1.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x0301	Thermostat		Simple	Endpoint

9.1.3. Conditions

See the Base Device Type definition for conformance tags.

9.1.4. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

ID	Cluster	Client/Server	Quality	Conformance
0x0003	Identify	Server		M

ID	Cluster	Client/Server	Quality	Conformance
0x0004	Groups	Server		Active
0x0201	Thermostat	Server		M
0x0204	Thermostat User Interface Configuration	Server		O
0x009B	Energy Preference	Server		O
0x0202	Fan Control	Client		O
0x0402	Temperature Measurement	Client		O
0x0405	Relative Humidity Measurement	Client		O
0x0406	Occupancy Sensing	Client		O

9.1.5. Element Requirements

Below list qualities and conformance that override the cluster specification requirements. A blank table cell means there is no change to that item and the value from the cluster specification applies.

ID	Cluster	Element	Name	Constraint	Access	Conformance
0x0201	Thermostat	Feature	ScheduleConfiguration			X
0x0201	Thermostat	Attribute	AlarmMask			X
0x0201	Thermostat	Command	GetRelayStatusLog			X
0x0201	Thermostat	Command	GetRelayStatusLogResponse			X

9.2. Fan

A Fan device is typically standalone or mounted on a ceiling or wall and is used to circulate air in a room.

9.2.1. Revision History

This is the revision history for this device type. The highest revision number in the table below is the revision for this device type.

Revision	Description
1	Initial revision
2	Added ability to be composed with a Thermostat for fan heaters
3	Added On/Off cluster

9.2.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x002B	Fan		Simple	Endpoint

9.2.3. Conditions

See the Base Device Type definition for conformance tags.

9.2.4. Device Type Requirements

A fan MAY expose elements of its functionality through one or more additional device types on different endpoints. All devices used in compositions SHALL adhere to the disambiguation requirements of the System Model. Other device types, not explicitly listed in the table, MAY also be included in device compositions but are not considered part of the core functionality of the device.

ID	Name	Constraint	Conformance
0x0301	Thermostat		O

9.2.5. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

ID	Cluster	Client/Server	Quality	Conformance
0x0003	Identify	server		M
0x0004	Groups	server		M
0x0006	On/Off	server		O
0x0202	Fan Control	server		M

9.2.6. Cluster Restrictions

9.2.6.1. On/Off Cluster (Server) Clarifications

The On/Off cluster is independent from the Fan Control Cluster’s FanMode attribute, which also includes an `Off` setting.

If the `FanMode` attribute of the Fan Control cluster is set to a value other than `Off` when the OnOff attribute of the On/Off cluster transitions from TRUE to FALSE, it may be desirable to restore the FanMode, SpeedSetting and PercentSetting attribute values of the Fan Control cluster when the OnOff attribute of the On/Off cluster later transitions from FALSE to TRUE. If the FanMode is set to Off when the device is turned off, this information is lost, as the SpeedSetting and PercentSetting will be set to zero. Using the On/Off cluster alongside the Fan Control cluster allows the FanMode, SpeedSetting and PercentSetting to remain unchanged when the device is turned off. In this case, the On/Off cluster would be set to Off, and the SpeedCurrent and PercentCurrent set to zero, without changing FanMode, SpeedSetting and PercentSetting.

9.2.7. Element Requirements

Below list qualities and conformance that override the cluster specification requirements. A blank entry means no change.

ID	Cluster	Element	Name	Quality	Constraint	Access	Conformance
0x0202	Fan Control	Attribute	FanModeSequence	F		R V	Matter

9.3. Air Purifier

An Air Purifier is a standalone device that is designed to clean the air in a room.

It is a device that has a fan to control the air speed while it is operating. Optionally, it can report on the condition of its filters.

9.3.1. Revision History

This is the revision history for this device type. The highest revision number in the table below is the revision for this device type.

Revision	Description
1	Initial revision
2	Added On/Off cluster

9.3.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x002D	Air Purifier		Simple	Endpoint

9.3.3. Conditions

See the Base Device Type definition for conformance tags.

9.3.4. Device Type Requirements

An Air Purifier MAY expose elements of its functionality through one or more additional device types on different endpoints. All devices used in compositions SHALL adhere to the disambiguation requirements of the System Model. Other device types, not explicitly listed in the table, MAY also be included in device compositions but are not considered part of the core functionality of the device.

ID	Name	Constraint	Conformance
0x0301	Thermostat		O
0x0302	Temperature Sensor		O
0x0307	Humidity Sensor		O
0x002C	Air Quality Sensor		O

9.3.5. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

ID	Cluster	Client/Server	Quality	Conformance
0x0003	Identify	Server		M
0x0004	Groups	Server		O
0x0006	On/Off	Server		O
0x0202	Fan Control	Server		M
0x0071	HEPA Filter Monitoring	Server		O
0x0072	Activated Carbon Filter Monitoring	Server		O

9.3.6. Cluster Restrictions

9.3.6.1. On/Off Cluster (Server) Clarifications

The On/Off cluster is independent from the Fan Control Cluster’s FanMode attribute, which also includes an `Off` setting.

If the `FanMode` attribute of the Fan Control cluster is set to a value other than `Off` when the OnOff attribute of the On/Off cluster transitions from TRUE to FALSE, it may be desirable to restore the `FanMode`, `SpeedSetting` and `PercentSetting` attribute values of the Fan Control cluster when the OnOff attribute of the On/Off cluster later transitions from FALSE to TRUE. If the `FanMode` is set to `Off` when the device is turned off, this information is lost, as the `SpeedSetting` and `PercentSetting` will be set to zero. Using the On/Off cluster alongside the Fan Control cluster allows the `FanMode`, `SpeedSetting` and `PercentSetting` to remain unchanged when the device is turned off. In this case, the On/Off cluster would be set to `Off`, and the `SpeedCurrent` and `PercentCurrent` set to zero, without changing `FanMode`, `SpeedSetting` and `PercentSetting`.

9.3.7. Element Requirements

There are no cluster element overrides.

9.4. Thermostat Controller Device Type

A Thermostat Controller is a device capable of controlling a Thermostat.

9.4.1. Revision History

Revision	Description
1	Initial Matter revision

9.4.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x030A	Thermostat Controller		Simple	Endpoint

9.4.3. Conditions

Please see the Base Device Type definition for conformance tags.

9.4.4. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

Table 21. Thermostat Controller Cluster Requirements

ID	Name	Client/Server	Quality	Conformance
0x0003	Identify	Client		O
0x0004	Groups	Client		O
0x0062	Scenes Management	Client		O
0x0201	Thermostat	Client		M

Chapter 10. Media Device Types

10.1. Video Player Architecture

10.1.1. Introduction

A Video Player endpoint (either Casting Video Player or Basic Video Player) represents a device that is able to play media to a physical output or to a display screen which is part of the device. For example, a Video Player can be a traditional TV device, a physical media playback device such as a DVD Player, a TV Set Top Box, or a content streaming device that provides input to another device like a TV or computer monitor.

Video Player features can be categorized into **basic** and **content launching**.

The **basic** features include (conceptually): On/Off, Volume Control, Playback Control, Channel Change, Input Control, Output Control, Sleep/Wake, Target Navigation and Keypad Navigation.

The **content launching** features include: discovery and launch of Content Apps, search and launch of content by content name and by URL.

A Basic Video Player is a **commissionable node** and supports these basic features which include, at a minimum, media playback controls (Media Playback cluster server) and remote controls (Keypad Input cluster server).

A Casting Video Player is a **commissioner** and supports both the Basic Video Player features and content launching features which include, at a minimum, the ability to launch content (Content Launcher cluster server).

A Content App is usually an application built by a Content Provider and exists as a separate endpoint on a Casting Video Player with a Content App Platform.

When a Casting Video Player includes a Content App Platform, it can launch Content Apps (Application Launcher cluster server) and represent these apps as separate endpoints on the Node.

A Video Remote Control is a **commissionable node** used to control basic features including, at a minimum, the ability to initiate keypad navigation (Keypad Input cluster client) and media playback (Media Playback cluster client).

A Casting Video Client is a **commissionable node** which extends the Video Remote Control features with the ability to initiate content launching (Content Launcher cluster client). A Casting Video Client is often associated with a Content App built by a specific Content Provider - for example, the Vendor Id of the Content App's Application Basic cluster will match the Vendor Id of the Casting Video Client.

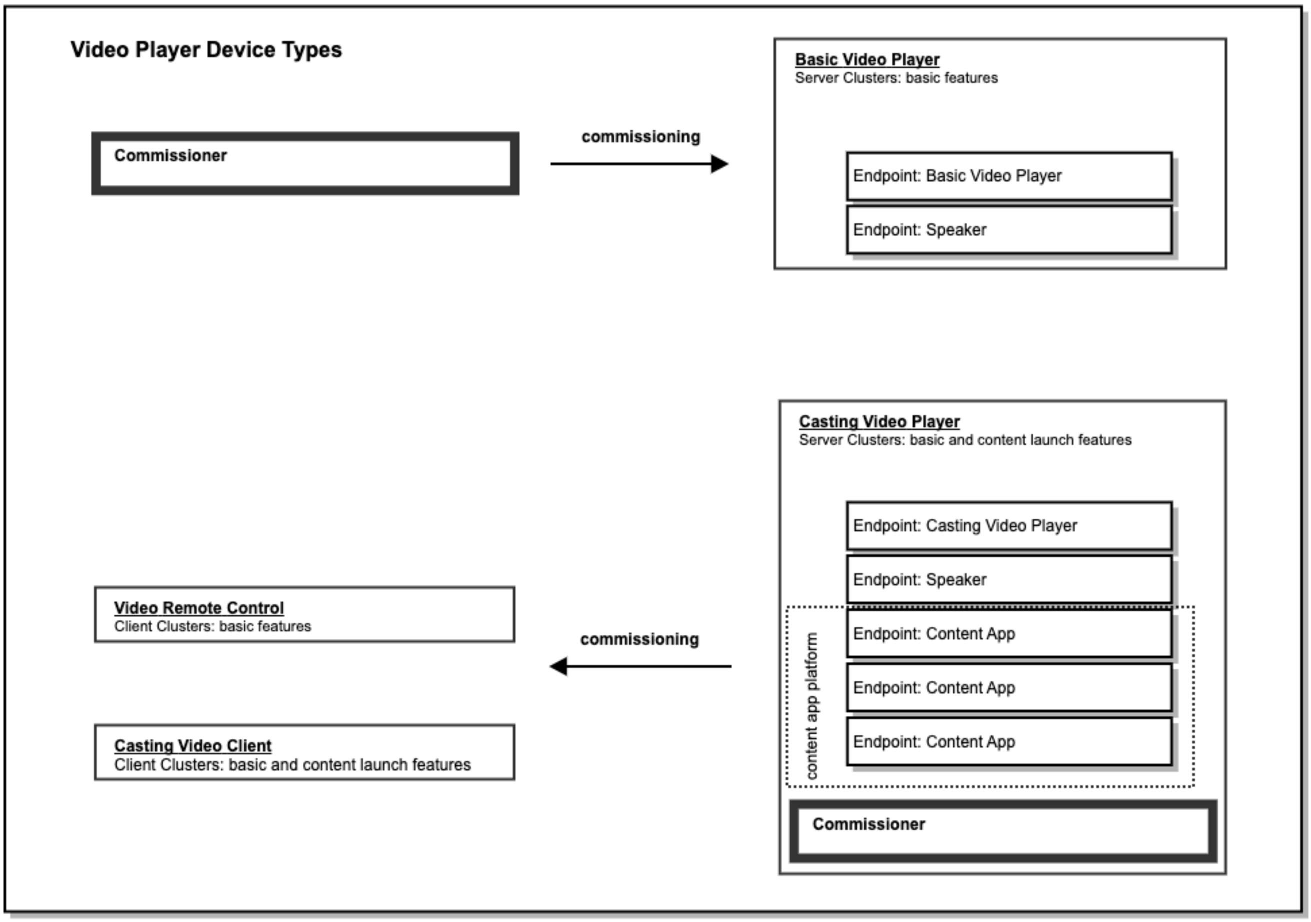

Video Player Device Types
Commissioner
commissioning
Basic Video Player
Server Clusters: basic features
Endpoint: Basic Video Player
Endpoint: Speaker
Casting Video Player
Server Clusters: basic and content launch features
Endpoint: Casting Video Player
Endpoint: Speaker
content app platform
Endpoint: Content App
Endpoint: Content App
Endpoint: Content App
Commissioner
Video Remote Control
Client Clusters: basic features
commissioning
Casting Video Client
Client Clusters: basic and content launch features

Figure 2. Video Player Device Types

10.1.2. Clients of a Casting Video Player

The clients for a Video Player device can be categorized into 2 high-level groups.

1. Clients controlling the Video Player endpoint, such as a remote control (e.g.: Video Remote device type)
2. Clients controlling specific Content App(s) such as a Phone App casting to a corresponding Content App (e.g.: Casting Video Client device type)

10.1.3. Endpoint Composition for Content Apps of a Casting Video Player

A Casting Video Player with a Content App Platform SHALL represent each Content App as its own dedicated endpoint where each is identified using the Device ID 0x0024 for "Content App".

The requirements for allocating and deallocating an endpoint address for a Content App SHALL be as described in the System Model specification (see "Dynamic Endpoint allocation").

The following diagram shows a Video Player device containing 3 separate Content Apps:

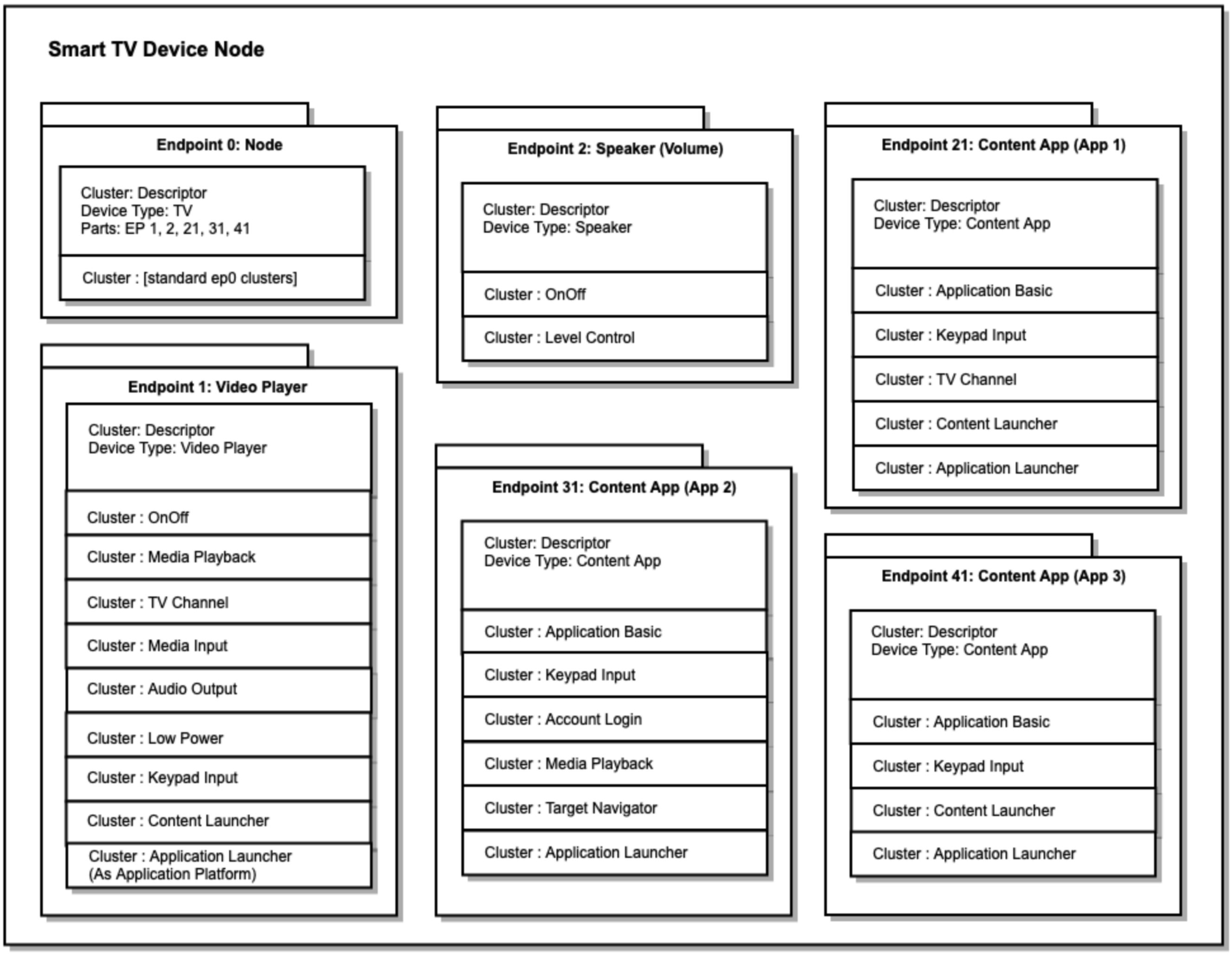


Figure 3. Endpoint Composition for Video Player Device

10.1.4. Commissioning

A Basic Video Player SHALL be commissioned like any commissionable node.

A Casting Video Player SHALL be a Commissioner. The primary reason for this requirement is to enable the Casting Video Player to verify, using device attestation, the vendor of a Client for the purpose of controlling content on the Casting Video Player, and to ensure that only clients authorized by a Content App can control it. In this way, a Commissionee associated with a Content App on the Casting Video Player can be commissioned by the Casting Video Player and granted access to the endpoint associated with that Content App.

When a Casting Video Player commissions a Client, such as a Video Remote Control or a Casting Video Client, the Casting Video Player SHALL determine the Content App access for the given Client following the rules defined in this section. Since the Client is being commissioned by the Casting Video Player, the Client, which may be a phone app, SHALL include attestation credentials which are used by the Casting Video Player to determine its Vendor ID.

1. A Casting Video Player SHOULD allow each Content App to specify which clusters it implements, and reflect these clusters in the corresponding endpoint for the given Content App. The method for conveying this information between the Content App and the Casting Video Player is specific to the vendor of the Casting Video Player.

2. A Casting Video Player SHALL allow each Content App to specify values for fields in the Application Basic cluster, such as Vendor ID and Application Name. A Casting Video Player SHALL also use this information to determine access control for Clients commissioned by the Casting Video Player. The method for conveying this information between the Content App and the Casting Video Player is specific to the vendor of the Casting Video Player.
3. A Casting Video Player device SHALL allow each Content App to provide an Allowed Controller Vendor ID list. The Allowed Controller Vendor ID list specifies a list of Vendor IDs for Clients that SHALL be granted access to the endpoint for the given Content App. The Casting Video Player device SHALL use the Allowed Controller Vendor ID list to determine access control for Clients commissioned by the Casting Video Player. Only Clients with a Vendor ID in the Allowed Controller Vendor ID list SHALL be granted access to the given Content App endpoint. The method for conveying this information between the Content App and the Casting Video Player is specific to the vendor of the Casting Video Player. When a Client is commissioned, its attested Vendor ID is used to determine access to Content App endpoints. The Allowed Controller Vendor ID list is contained in the AllowedVendorList attribute of the Application Basic cluster.

A Casting Video Player MAY enable a commissioning flow, which avoids Setup PIN entry by the user, when the following conditions are met:

1. The Client’s Vendor ID and a Rotating ID (used as a TempAccountIdentifier) are present in its commissionable node advertisement, or present in a User Directed Commissioning message sent by the Client to initiate commissioning with the Casting Video Player.
2. The Casting Video Player is able to determine an endpoint corresponding to the given Vendor ID which contains the Account Login cluster (for example, a Content App endpoint).
3. The Account Login cluster’s GetSetupPIN command returns a SetupPIN which is then used successfully to commission the Client.

A Casting Video Player MAY enable a Content App account login flow, which avoids login name and password entry by the user, when the following conditions are met:

1. The Client’s Vendor ID and a Rotating ID (used as a TempAccountIdentifier) are present in its commissionable node advertisement, or present in a User Directed Commissioning message sent by the Client to initiate commissioning with the Casting Video Player.
2. The Casting Video Player is able to determine an endpoint corresponding to the given Vendor ID which contains the Account Login cluster (for example, a Content App endpoint).
3. The Account Login cluster’s Login command returns successfully.

In these flows, when the Account Login cluster is located on a Content App endpoint, the Account Login cluster will often be implemented by a different vendor from the Casting Video Player itself. See AccountLoginCluster for further details on the use of these commands.

Since a Client commissioned by the Casting Video Player will only have access to one or more Content App endpoints and the Speaker endpoint (when present), it will not have the ability to access the Application Launcher cluster on the Casting Video Player endpoint. If they do not already exist, the Casting Video Player device SHALL create endpoints for each Content App to which the Client has access and notify the Client of such access by adding an entry for each Content App to the Binding cluster of the Client. The Casting Video Player device SHALL automatically launch the Content

App upon commands targeted to a Content App endpoint.

The following steps are performed by a Casting Video Player when granting and removing Client access to Content App endpoints:

1. Upon commissioning a Client and granting it access to a Content App endpoint, the Casting Video Player SHALL invoke the Login command of the AccountLogin cluster on the Content App endpoint (when implemented by the Content App) in order to notify the Content App endpoint of the Client.
    a. If the Client is a Casting Video Client and it implements the Content App Observer server cluster, and if the Content App endpoint implements both the Binding server cluster and the Content App Observer client cluster, then the Casting Video Player SHALL create a binding on the Content App endpoint which specifies the Node and Endpoint of the Casting Video Client's Content App Observer server cluster. The Casting Video Player SHALL provide a way for the Content App to send a Content App Observer cluster ContentAppMessage command to the Casting Video Client and to receive a ContentAppMessageResponse in return.
2. Upon removing access to a Content App endpoint for a Client, the Casting Video Player SHALL invoke the Logout command of the AccountLogin cluster on the Content App endpoint (when implemented by the Content App) in order to notify the Content App endpoint that Client access has been removed.
    a. If the Casting Video Player created a binding on the Content App endpoint corresponding to the Casting Video Client's Content App Observer server cluster, then the Casting Video Player SHALL remove this binding.

When the Content App endpoint generates a LoggedOut event, the Casting Video Player SHALL remove access to the Node specified to the given Content App endpoint.

NOTE A Client commissioned by the Casting Video Player is able to determine if the corresponding Content App is visible to the user using the **Status** attribute on the Application Basic cluster for the Content App endpoint. This ensures that the Client cannot access foreground Content App information about any other Content App to which it does not have access. It also ensures that such a client will only need access to specific Content App endpoints and the Speaker endpoint (when present).

10.1.5. Determining Context

A client that controls multiple aspects of the Video Player functionality (like a voice assistant) may have access to multiple endpoints on the Video Player. To determine the current context on the Video Player when the user interacts with the client (for example, when the user interacts with the device using voice), the client can look at the state in the various clusters on the Casting Video Player.

Specifically:

1. Media Input cluster (when CurrentInput does NOT have type INTERNAL, then Video Player is displaying content from a physical input)
2. Application Launcher cluster (CurrentApp indicates the current application endpoint - which

may be the Video Player endpoint when no Content App is in the foreground).

3. Target Navigator cluster on current application endpoint (indicates which navigation target the user is in)

The Video Player SHOULD provide a way for the user to view the list of clients with access to control the screen and SHOULD provide a way for the user to revoke this access.

10.1.6. Basic Video Player Features

10.1.6.1. On/Off/Toggle

This feature turns on/off the user-visible power state of the device, corresponding to the on/off/toggle button usually found on a remote or button on the device.

An On/Off cluster on the Video Player endpoint SHALL be used for this feature.

10.1.6.2. Volume Control

This feature controls the speaker volume of the device.

A Speaker endpoint SHALL be used for this feature when the device controls a speaker.

10.1.6.3. Media Playback Control

This feature controls media playback on the device which includes functionality such as Play, Pause, Stop, Rewind, and Fast Forward.

The Media Playback cluster SHALL be used for this functionality.

10.1.6.4. Channel Change

This feature controls channel control functionality on the device which includes functionality such as lineup discovery, change and skip channels.

The Channel cluster SHALL be used for this functionality.

10.1.6.5. Media Input Control

This feature controls the input selection on the device which includes functionality such as input discovery, selection and naming.

The Media Input cluster SHALL be used for this functionality.

10.1.6.6. Audio Output Control

This feature controls audio output selection on the device which includes functionality such as output discovery, selection and naming.

The Audio Output cluster SHALL be used for this functionality.

Note that when the current output is set to an output of type HDMI, adjustments to volume via a

Speaker endpoint on the same node MAY cause HDMI volume up/down commands to be sent to the given HDMI output.

10.1.6.7. Sleep/Wake

This feature controls low power mode on the device which includes functionality such as sleep, and declaration of protocols supported for Wakeup.

The Low Power cluster SHALL be used for putting a device into low power (sleep) mode.

The WakeOnLAN cluster SHALL be used for declaring that a device supports the WakeOnLAN protocol.

10.1.6.8. Target Navigation

This feature controls on-screen navigation to custom-named targets, for example, "Settings", "On Demand" and "Search". A list of named targets can be provided for the Video Player endpoint itself, as well as for Content App represented as endpoints.

The Target Navigator cluster SHALL be used for listing navigation targets, invoking navigation to a target, and tracking the current target.

10.1.6.9. Keypad Navigation

This feature controls on-screen navigation, commonly referred to as D-Pad navigation, and includes navigation commands such as UP, DOWN, LEFT, RIGHT, SELECT, BACK, EXIT, MENU.

The Keypad Input cluster SHALL be used for this functionality.

10.1.6.10. Account Login

This cluster provides commands that facilitate user account login on an application or a node.

The Account Login cluster SHALL be used for this functionality.

10.1.7. Content Launching Features

Many Video Player devices (traditional TVs, Set Top Boxes, Content Streamers) have advanced features that MAY include any of the following:

- the ability to search for and playback content such as movies and TV shows
- a platform for Content Apps that can themselves be launched, and instructed to search and playback content such as movies and TV shows
- the ability to download and playback basic content referenced by URL

10.1.7.1. Discover and Launch Content App from another Device

This feature allows a client to discover the Content App identification catalogs supported by a Video Player device, and launch an Application based upon a Content App identifier within a given catalog. An example Content App identification catalog is the DIAL registry (http://www.dialmultiscreen.org/).

The Application Launcher cluster SHALL be used for this functionality.

10.1.7.2. Launch Content from another Device

Content search and launch is defined by the Content Launcher cluster which includes feature flags for Content Search and URL Playback which are used to indicate which of these features is supported.

The Content Launcher cluster SHALL be used for this functionality.

10.2. Basic Video Player

This defines conformance to the Basic Video Player device type.

A Video Player (either Basic or Casting) represents a device that is able to play media to a physical output or to a display screen which is part of the device.

A Basic Video Player has playback controls (play, pause, etc.) and keypad remote controls (up, down, number input), but is not able to launch content and is not a content app platform (the Casting Video Player device type is used for these functions).

For example, a Basic Video Player can be a traditional TV device a physical media playback device such as a DVD Player, or a device that provides input to another device like a TV or computer monitor.

See Section 10.1, “Video Player Architecture” for additional Basic Video Player requirements relating to Video Player device endpoint composition, commissioning, feature representation in clusters, and UI context.

10.2.1. Revision History

This is the revision history for this device type. The highest revision number in the table below is the revision for this device type.

Revision	Description
1	Initial revision
2	Added Messages and Content Control clusters

10.2.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x0028	Basic Video Player		Simple	Endpoint

10.2.3. Conditions

This device type SHALL support the following conformance conditions as defined below.

Feature	Description
PhysicalInputs	The device has physical inputs for media.

See the Base Device Type definition for additional conformance tags.

10.2.4. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

ID	Cluster	Client/Server	Quality	Conformance
0x0006	On/Off	Server		M
0x0503	Wake On LAN	Server		O
0x0504	Channel	Server		O
0x0505	Target Navigator	Server		O
0x0506	Media Playback	Server		M
0x0507	Media Input	Server		PhysicalInputs
0x0508	Low Power	Server		O
0x0509	Keypad Input	Server		M
0x050B	Audio Output	Server		O
0x050F	Content Control	Server		P, O
0x0097	Messages	Server		O

10.3. Casting Video Player

This defines conformance to the Casting Video Player device type.

A Video Player (either Basic or Casting) represents a device that is able to play media to a physical output or to a display screen which is part of the device.

A Casting Video Player has basic controls for playback (play, pause, etc.) and keypad input (up, down, number input), and is able to launch content.

For example, a Casting Video Player can be a smart TV device, a TV Set Top Box, or a content streaming device that provides input to another device like a TV or computer monitor.

See Section 10.1, “Video Player Architecture” for additional Casting Video Player requirements relating to Video Player device endpoint composition, commissioning, feature representation in clusters, and UI context.

10.3.1. Revision History

This is the revision history for this device type. The highest revision number in the table below is the revision for this device type.

Revision	Description
1	Initial revision
2	Added Messages and Content Control clusters

10.3.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x0023	Casting Video Player		Simple	Endpoint

10.3.3. Conditions

This device type SHALL support the following conformance conditions as defined below.

Feature	Description
ContentAppPlatform	The device includes a Content App Platform. A Content App is usually an application built by a Content Provider. A Casting Video Player with a Content App Platform is able to launch Content Apps and represent these apps as separate endpoints.
PhysicalInputs	The device has physical inputs for media.

See the Base Device Type definition for additional conformance tags.

10.3.4. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

ID	Cluster	Client/Server	Quality	Conformance
0x0006	On/Off	Server		M
0x0503	Wake On LAN	Server		O
0x0504	Channel	Server		O
0x0505	Target Navigator	Server		O
0x0506	Media Playback	Server		M
0x0507	Media Input	Server		PhysicalInputs
0x0508	Low Power	Server		O
0x0509	Keypad Input	Server		M
0x050A	Content Launcher	Server		M
0x050B	Audio Output	Server		O

ID	Cluster	Client/Server	Quality	Conformance
0x050C	Application Launcher	Server		ContentAppPlatform
0x050E	Account Login	Server		O
0x050F	Content Control	Server		P, O
0x0097	Messages	Server		O

10.3.5. Element Requirements

Below list qualities and conformance that override the cluster specification requirements. A blank entry means no change.

ID	Cluster	Element	Name	Constraint	Access	Conformance
0x050C	Application Launcher	Feature	ApplicationPlatform			M

10.4. Speaker

This defines conformance to the Speaker device type.

This feature controls the speaker volume of the device.

To control unmute/mute, the On/Off cluster SHALL be used. A value of TRUE for the OnOff attribute SHALL represent the volume on (not muted) state, while a value of FALSE SHALL represent the volume off (muted) state. For volume level control, the Level cluster SHALL be used.

A dedicated endpoint is needed because the On/Off cluster can also be used for other purposes, such as for power control.

The decision to use Level and On/Off clusters for volume (rather than defining a new audio control cluster) was made in order to treat volume in a fashion consistent with lighting which also uses these clusters and has matching functional requirements.

10.4.1. Revision History

This is the revision history for this device type. The highest revision number in the table below is the revision for this device type.

Revision	Description
1	Initial revision

10.4.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x0022	Speaker		Simple	Endpoint

10.4.3. Conditions

See the Base Device Type definition for conformance tags.

10.4.4. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

ID	Cluster	Client/Server	Quality	Conformance
0x0006	On/Off	Server		M
0x0008	Level Control	Server		M

10.5. Content App

This defines conformance to the Content App device type.

A Content App is usually an application built by a Content Provider. A Casting Video Player with a Content App Platform is able to launch Content Apps and represent these apps as separate endpoints.

10.5.1. Revision History

This is the revision history for this device type. The highest revision number in the table below is the revision for this device type.

Revision	Description
1	Initial revision
2	Added Content App Observer cluster

10.5.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x0024	Content App		Simple	Endpoint

10.5.3. Conditions

Condition	Description
ObserverClient	The node is a client for ContentAppObservers.

See the Base Device Type definition for additional conformance tags.

10.5.4. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

ID	Cluster	Client/Server	Quality	Conformance
0x001E	Binding	Server		ObserverClient
0x0504	Channel	Server		O
0x0505	Target Navigator	Server		O
0x0506	Media Playback	Server		O
0x0509	Keypad Input	Server		M
0x050A	Content Launcher	Server		O
0x050C	Application Launcher	Server		M
0x050D	Application Basic	Server		M
0x050E	Account Login	Server		O
0x0510	Content App Observer	Client		ObserverClient

10.5.5. Element Requirements

Below list qualities and conformance that override the cluster specification requirements. A blank entry means no change.

ID	Cluster	Element	Name	Constraint	Access	Conformance
0x050C	Application Launcher	Feature	Application-Platform			X

10.5.6. Endpoint Composition

Endpoints with this device type SHALL support Dynamic Endpoint Allocation as specified in the System Model specification.

10.5.7. Disambiguation

When there is more than one sibling endpoint with this device type in a PartsList, disambiguation information SHALL be provided by having a unique value in the ApplicationName attribute of the Application Basic cluster on each of these endpoints.

10.6. Casting Video Client

This defines conformance to the Casting Video Client device type.

A Casting Video Client is a client that can launch content on a Casting Video Player, for example, a Smart Speaker or a Content Provider phone app.

10.6.1. Revision History

This is the revision history for this device type. The highest revision number in the table below is the revision for this device type.

Revision	Description
1	Initial revision
2	Added Content App Observer, Messages and Content Control Clusters

10.6.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x0029	Casting Video Client		Simple	Endpoint

10.6.3. Conditions

See the Base Device Type definition for additional conformance tags.

10.6.4. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

See Section 1.1.7, “Cluster Requirements” for additional clusters including the Binding cluster.

ID	Cluster	Client/Server	Quality	Conformance
0x0006	On/Off	Client		M
0x0008	Level Control	Client		O
0x0503	Wake On LAN	Client		O
0x0504	Channel	Client		O
0x0505	Target Navigator	Client		O
0x0506	Media Playback	Client		O
0x0507	Media Input	Client		O
0x0508	Low Power	Client		O
0x0509	Keypad Input	Client		M
0x050A	Content Launcher	Client		M
0x050B	Audio Output	Client		O

ID	Cluster	Client/Server	Quality	Conformance
0x050C	Application Launcher	Client		O
0x050D	Application Basic	Client		M
0x050E	Account Login	Client		O
0x050F	Content Control	Client		P, O
0x0510	Content App Observer	Server		O
0x0097	Messages	Client		O

10.7. Video Remote Control

This defines conformance to the Video Remote Control device type.

A Video Remote Control is a client that can control a Video Player, for example, a traditional universal remote control.

10.7.1. Revision History

This is the revision history for this device type. The highest revision number in the table below is the revision for this device type.

Revision	Description
1	Initial revision
2	Added Content Control cluster

10.7.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x002A	Video Remote Control		Simple	Endpoint

10.7.3. Conditions

See the Base Device Type definition for additional conformance tags.

10.7.4. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

ID	Cluster	Client/Server	Quality	Conformance
0x0006	On/Off	Client		M

ID	Cluster	Client/Server	Quality	Conformance
0x0008	Level Control	Client		O
0x0503	Wake On LAN	Client		O
0x0504	Channel	Client		O
0x0505	Target Navigator	Client		O
0x0506	Media Playback	Client		M
0x0507	Media Input	Client		O
0x0508	Low Power	Client		O
0x0509	Keypad Input	Client		M
0x050A	Content Launcher	Client		O
0x050B	Audio Output	Client		O
0x050C	Application Launcher	Client		O
0x050E	Account Login	Client		O
0x050F	Content Control	Client		P, O

Chapter 11. Generic Device Types

11.1. Mode Select

This defines conformance to the Mode Select device type.

11.1.1. Revision History

This is the revision history for this device type. The highest revision number in the table below is the revision for this device type.

Revision	Description
1	Initial revision

11.1.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x0027	Mode Select		Simple	Endpoint

11.1.3. Conditions

See the Base Device Type definition for conformance tags.

11.1.4. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

ID	Cluster	Client/Server	Quality	Conformance
0x0050	Mode Select	Server		M

11.2. Aggregator

This device type aggregates endpoints as a collection. Clusters on the endpoint indicating this device type provide functionality for the collection of descendant endpoints present in the PartsList of the endpoint's descriptor, for example the Actions cluster.

The purpose of this device type is to aggregate functionality for a collection of endpoints. The definition of the collection or functionality is not defined here.

When using this device type as a collection of bridged nodes, please see the "Bridge" section in the System Model specification.

11.2.1. Revision History

This is the revision history for this device type. The highest revision number in the table below is the revision for this device type.

Revision	Description
1	Initial revision
2	Added Commissioner Control Cluster and Fabric-Synchronization Condition.

11.2.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x000E	Aggregator		Simple	Endpoint

11.2.3. Conditions

This device type MAY support the following conformance conditions as defined below.

Condition	Description
FabricSynchronization	See description below.

See the Base Device Type definition for additional conformance tags.

11.2.3.1. FabricSynchronization Condition

The FabricSynchronization condition applies when there is a Commissioner Control Cluster on this endpoint with a SupportedDeviceCategories attribute with the FabricSynchronization bit set.

11.2.4. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

ID	Cluster	Client/Server	Quality	Conformance
0x0025	Actions	Server		O
0x0003	Identify	Server		O
0x0751	Commissioner Control	Server		FabricSynchronization

The Identify cluster SHOULD be used in case this device type is used to represent a Bridge which has a mechanism to identify itself to the user (e.g. blinking LED on the bridge itself).

For the Identify-functionality of the individual bridged devices, see the Identify cluster on the endpoint for a bridged device.

11.2.5. Endpoint Composition

An Aggregator endpoint's Descriptor cluster PartsList attribute SHALL list the collection of all endpoints aggregated by the Aggregator device type, i.e. the full-family pattern defined in the System Model specification.

11.2.5.1. Multiple aggregators

When a Node has multiple instances of the Aggregator device type, the composition SHALL comply with one of the following two patterns for any given pair (A,B) of endpoints with the Aggregator device type:

- **No overlap**: The endpoints in the PartsList attribute of Aggregator A do not appear in the PartsList attribute of Aggregator B, and vice versa.
  - Example: A Node which bridges to two non-Matter independent technologies (e.g. Zigbee and Z-Wave), see the aggregators on endpoints 11 and 31 in the figure below - their lists of endpoints (12-14, 21-23 versus 32-33) do not overlap.
- **Strict subset**: The endpoint where aggregator B is exposed and all endpoints in its PartsList attribute (the subset) are included in the PartsList attribute of Aggregator A (the superset).
  - This maintains the rule that there SHALL be a single path from the Root Node to each endpoint (see System Model).
  - Example: A Node which implements a bridge to Zigbee, and one of those Zigbee devices is connected to a string of DALI lights, which can be addressed individually and thus this Zigbee/DALI device functions as a bridge from Zigbee to DALI; in the figure below one can see that the endpoints for the Zigbee/DALI bridge listed in the PartsList of the aggregator on endpoint 14 (21-23) form a strict subset of the endpoints for the Zigbee bridge in the PartsList of the aggregator on endpoint 11 (12-14, 21-23), and the endpoint 14 of the "subset" aggregator is included in the PartsList of the "superset" aggregator on endpoint 11.

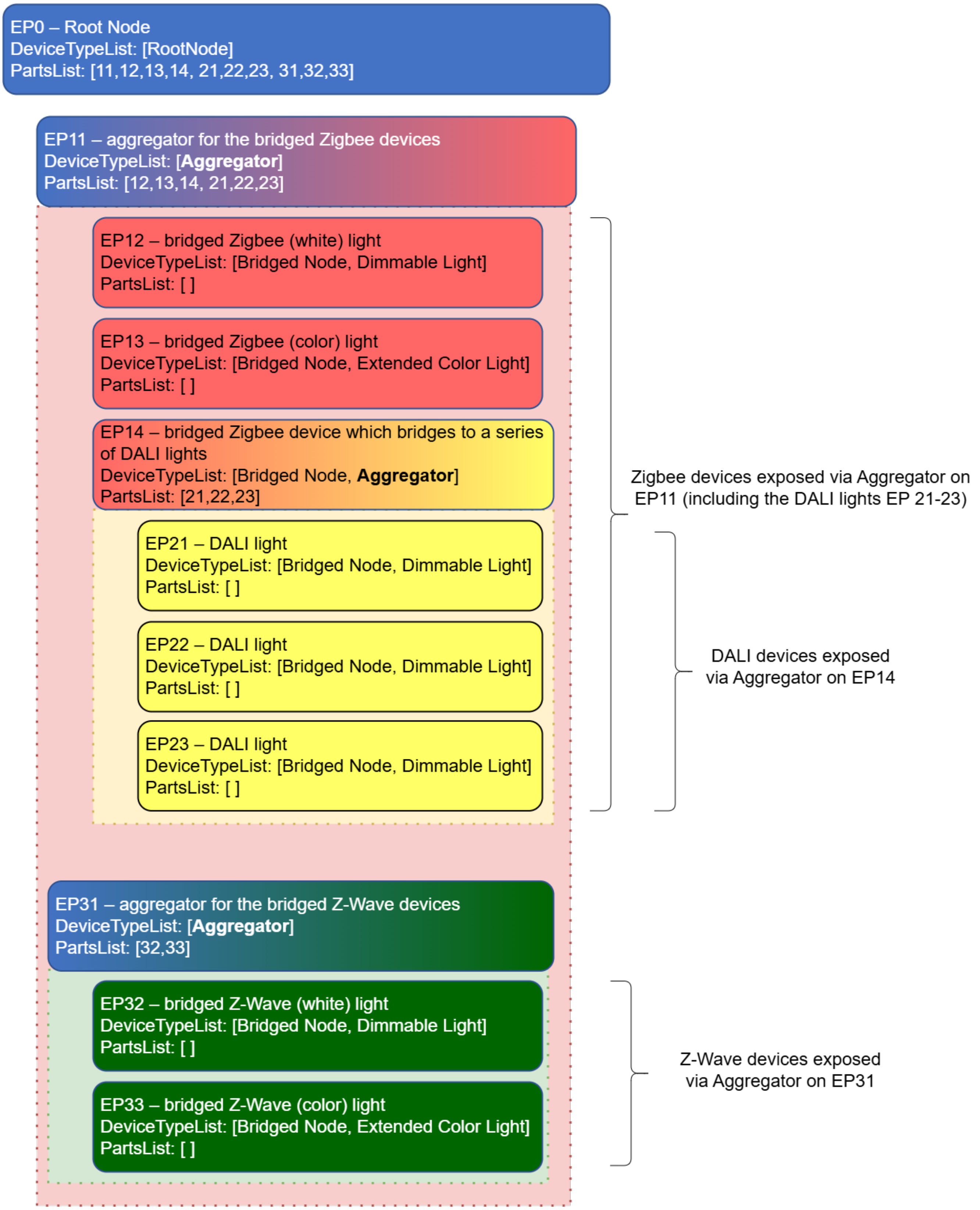

EP0 – Root Node
DeviceTypeList: [RootNode]
PartsList: [11,12,13,14, 21,22,23, 31,32,33]
EP11 – aggregator for the bridged Zigbee devices
DeviceTypeList: [**Aggregator**]
PartsList: [12,13,14, 21,22,23]
EP12 – bridged Zigbee (white) light
DeviceTypeList: [Bridged Node, Dimmable Light]
PartsList: [ ]
EP13 – bridged Zigbee (color) light
DeviceTypeList: [Bridged Node, Extended Color Light]
PartsList: [ ]
EP14 – bridged Zigbee device which bridges to a series of DALI lights
DeviceTypeList: [Bridged Node, **Aggregator**]
PartsList: [21,22,23]
EP21 – DALI light
DeviceTypeList: [Bridged Node, Dimmable Light]
PartsList: [ ]
EP22 – DALI light
DeviceTypeList: [Bridged Node, Dimmable Light]
PartsList: [ ]
EP23 – DALI light
DeviceTypeList: [Bridged Node, Dimmable Light]
PartsList: [ ]
Zigbee devices exposed via Aggregator on EP11 (including the DALI lights EP 21-23)
DALI devices exposed via Aggregator on EP14
EP31 – aggregator for the bridged Z-Wave devices
DeviceTypeList: [**Aggregator**]
PartsList: [32,33]
EP32 – bridged Z-Wave (white) light
DeviceTypeList: [Bridged Node, Dimmable Light]
PartsList: [ ]
EP33 – bridged Z-Wave (color) light
DeviceTypeList: [Bridged Node, Extended Color Light]
PartsList: [ ]
Z-Wave devices exposed via Aggregator on EP31

Figure 4. examples of multiple aggregators

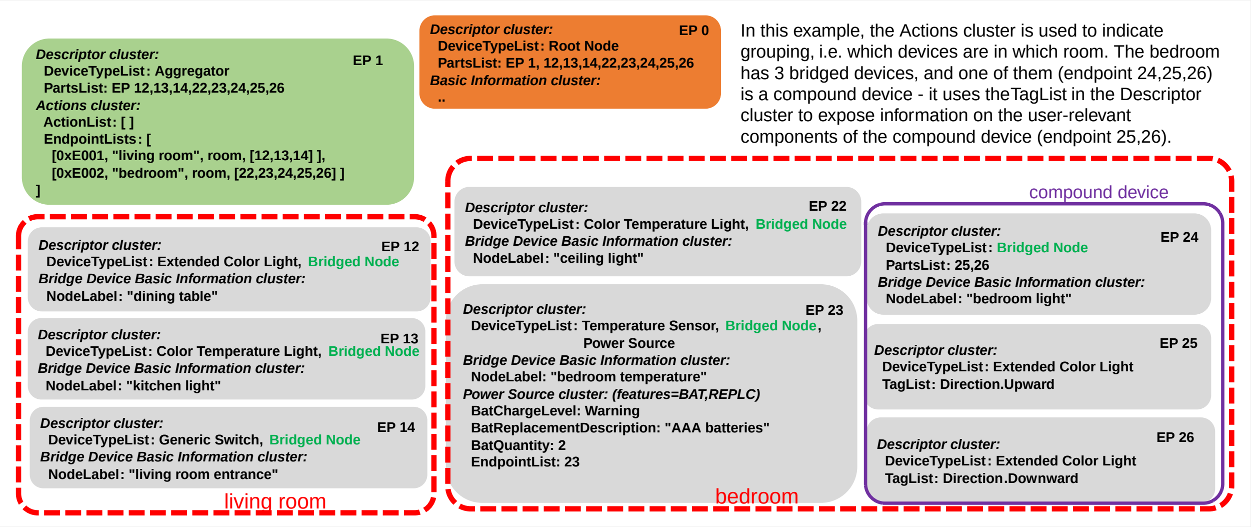
11.2.6. Disambiguation

If the Duplicate condition applies to child endpoints of an Aggregator endpoint that represent multiple independent bridged devices, the endpoints SHOULD make available metadata to allow a client to disambiguate distinct bridged devices with an overlap in application device types.

Typically this is done using the NodeLabel attribute of the Bridged Device Basic Information cluster

- thus reusing the naming information which the bridge already has to allow disambiguation to the user when using a direct user interface to the bridge.

Example: the Aggregator in this figure (copied from the "Bridge for non-Matter devices" section in the Core Specification) exposes several Color Temperature Lights (endpoints 13 and 22) which are disambiguated with their NodeLabel. Note that the compound device at endpoints 24, 25 and 26 also uses a TagList (for information rather than disambiguation) since, for this case, the bridge knows the lighting direction of both elements of the compound device.


Descriptor cluster:
DeviceTypeList: Aggregator
PartsList: EP 12,13,14,22,23,24,25,26
Actions cluster:
ActionList: [ ]
EndpointLists: [
[0xE001, "living room", room, [12,13,14]],
[0xE002, "bedroom", room, [22,23,24,25,26]]
]
EP 1
Descriptor cluster:
DeviceTypeList: Root Node
PartsList: EP 1, 12,13,14,22,23,24,25,26
Basic Information cluster:
..
EP 0
Descriptor cluster:
DeviceTypeList: Extended Color Light, Bridged Node
Bridge Device Basic Information cluster:
NodeLabel: "dining table"
EP 12
Descriptor cluster:
DeviceTypeList: Color Temperature Light, Bridged Node
Bridge Device Basic Information cluster:
NodeLabel: "kitchen light"
EP 13
Descriptor cluster:
DeviceTypeList: Generic Switch, Bridged Node
Bridge Device Basic Information cluster:
NodeLabel: "living room entrance"
EP 14
living room
Descriptor cluster:
DeviceTypeList: Color Temperature Light, Bridged Node
Bridge Device Basic Information cluster:
NodeLabel: "ceiling light"
EP 22
Descriptor cluster:
DeviceTypeList: Temperature Sensor, Bridged Node, Power Source
Bridge Device Basic Information cluster:
NodeLabel: "bedroom temperature"
Power Source cluster: (features=BAT,REPLC)
BatChargeLevel: Warning
BatReplacementDescription: "AAA batteries"
BatQuantity: 2
EndpointList: 23
EP 23
bedroom
compound device
Descriptor cluster:
DeviceTypeList: Bridged Node
PartsList: 25,26
Bridge Device Basic Information cluster:
NodeLabel: "bedroom light"
EP 24
Descriptor cluster:
DeviceTypeList: Extended Color Light
TagList: Direction.Upward
EP 25
Descriptor cluster:
DeviceTypeList: Extended Color Light
TagList: Direction.Downward
EP 26

In this example, the Actions cluster is used to indicate grouping, i.e. which devices are in which room. The bedroom has 3 bridged devices, and one of them (endpoint 24,25,26) is a compound device - it uses theTagList in the Descriptor cluster to expose information on the user-relevant components of the compound device (endpoint 25,26).

Figure 5. use of NodeLabel and TagList for disambiguation

Chapter 12. Robotic Device Types

12.1. Robotic Vacuum Cleaner Device Type

This defines conformance for the Robotic Vacuum Cleaner device type.

12.1.1. Revision History

This is the revision history for this document.

Revision	Description
1	Initial revision
2	Add cluster usage constraints and informative data. Remove the element requirements section, after moving all constraints to the respective cluster specifications.
3	Add support for the Service Area cluster
4	Mandate OperationCompletion Event

12.1.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x0074	Robotic Vacuum Cleaner		Simple	Endpoint

12.1.3. Conditions

See the Base Device Type definition for conformance tags.

12.1.4. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

ID	Cluster	Client/Server	Quality	Conformance
0x0003	Identify	Server		M
0x0054	RVC Run Mode	Server		M
0x0055	RVC Clean Mode	Server		O
0x0061	RVC Operational State	Server		M
0x0150	Service Area	Server		O

12.1.5. Element Requirements

Below list qualities and conformance that override the cluster specification requirements. A blank table cell means there is no change to that item and the value from the cluster specification applies.

ID	Cluster	Element	Name	Constraint	Access	Conformance
0x0061	RVC Operational State	Event	OperationCompletion			M

12.1.6. Cluster Usage

This section describes how to control and monitor the operation of a Robotic Vacuum Cleaner device. This information is meant to clarify how the data dependencies within the device type's cluster composition are to be used.

Note that the device operations may also be the result of, or affected by, out-of-band actions such as robot physical button presses, internally scheduled events, vendor application requests, commands sent from other fabrics, internal device timeouts, etc. For example, a user may pause the robot during cleaning by using a Matter client and then resume cleaning by using a physical button of the device, or a robot may stop cleaning after an internal timeout occurs, and so forth.

The RVC Operational State cluster's OperationalState attribute SHALL be updated according to the state of the device, and therefore it SHOULD be used for monitoring purposes. Note that while the robot is in a cleaning cycle it may automatically seek the charger, recharge, and then resume cleaning.

The sections below describe various operational flows with preconditions and actions. The behavior in case the preconditions are not met is described in the corresponding cluster descriptions.

12.1.6.1. Starting Cleaning

12.1.6.1.1. Preconditions

Cleaning can only be started when the RVC Run Mode cluster's CurrentMode attribute is set to a mode that has the Idle mode tag associated with it, and the RVC Operational State cluster's OperationalState attribute is set to the Stopped, Paused, Docked or Charging state.

Note that if the RVC Clean Mode cluster is implemented, it determines the type of cleaning.

12.1.6.1.2. Actions

To attempt starting a cleaning operation, the RVC Run Mode cluster can be sent a ChangeToMode command with the NewMode field set to a mode that has the Cleaning mode tag associated with it.

12.1.6.2. Pausing Cleaning

12.1.6.2.1. Preconditions

Cleaning can only be paused when the RVC Operational State cluster's OperationalState attribute is set to a Pause-compatible state. See the Pause Compatibility table and the RVC Pause Compatibility Table.

Note that even if the Pause command is not implemented, the RVC Operational State cluster's OperationalState attribute MAY report that the device is in the Paused state due to an out-of-band action, such as the user pressing a physical button on the device.

12.1.6.2.2. Actions

To attempt pausing a cleaning operation, the RVC Operational State cluster can be sent a Pause command.

12.1.6.3. Resuming Cleaning

12.1.6.3.1. Preconditions

Cleaning can only be resumed if the RVC Operational State cluster's OperationalState attribute is set to a Resume-compatible state (see Resume Compatibility table and the RVC Resume Compatibility table), and the RVC Run Mode cluster's CurrentMode is set to a mode with the Cleaning mode tag.

Note that even if the Resume command is not implemented, the RVC Operational State cluster's OperationalState attribute MAY indicate that the device transitioned from the Paused state to the Running state due to an out-of-band action, such as the user pressing a physical button on the device.

12.1.6.3.2. Actions

To attempt resuming a cleaning operation, the RVC Operational State cluster can be sent a Resume command.

12.1.6.4. Stopping Cleaning

12.1.6.4.1. Preconditions

Stopping cleaning can only happen if the RVC Run Mode cluster's CurrentMode attribute is set to a mode that has the Cleaning mode tag associated with it.

12.1.6.4.2. Actions

To attempt stopping a cleaning operation, the RVC Run Mode cluster can be sent a ChangeToMode command with the NewMode field set to a mode that has the Idle mode tag associated with it.

12.1.6.4.3. Side Effects

Note that the device MAY seek the charger after successfully switching the RVC Run Mode cluster to an Idle mode. The OperationalState attribute indicates whether the device is seeking the charger, stopped, charging, docked etc.

12.1.6.5. Other Device Operations

The RVC Run Mode cluster’s SupportedModes attribute list MAY include modes that have neither the Idle nor the Cleaning mode tags, for example the Mapping mode tag.

Starting, pausing, resuming and stopping these other operations have similar preconditions, actions and side effects as those described above for the cleaning operations.

12.1.6.6. Device Error Handling

When in an error condition, as indicated by the RVC Operational State cluster’s OperationalState attribute, out-of-band action will be required to clear that condition.

If an error occurs while the device operates, such as while cleaning or while mapping, the device MAY pause and set the RVC Operational State cluster’s OperationalState attribute to Error. If the operation can be resumed after the error is cleared, the device SHALL set the RVC Operational State cluster’s OperationalState attribute to Paused and MAY be resumed either via a Resume command, if implemented, or by out-of-band actions, such as by pressing the robot’s physical buttons.

Note that certain errors may not pause or disable the device. For example, a dual-function device, that can both vacuum and mop, may report a WaterTankEmpty error but may still be able to be used if it has a vacuum only cleaning mode. Certain modes of the RVC Run Mode and the RVC Cleaning Mode clusters MAY become unavailable and the ChangeToModeResponse commands' StatusCode SHALL be set to InvalidInMode, when attempting to switch to those modes.

Chapter 13. Appliances Device Types

13.1. Laundry Washer

A Laundry Washer represents a device that is capable of laundering consumer items. Any laundry washer product may utilize this device type.

A Laundry Washer SHALL be composed of at least one endpoint with the Laundry Washer device type.

13.1.1. Revision History

This is the revision history for this document.

Revision	Description
1	Initial revision
2	Mandate OperationCompletion Event

13.1.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x0073	Laundry Washer		Simple	Endpoint

13.1.3. Conditions

See the Base Device Type definition for conformance tags.

13.1.4. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

ID	Cluster	Client/Server	Quality	Conformance
0x0003	Identify	Server		O
0x0051	Laundry Washer Mode	Server		O
0x0006	On/Off	Server		O
0x0053	Laundry Washer Controls	Server		O
0x0056	Temperature Control	Server		O
0x0060	Operational State	Server		M

13.1.5. Cluster Restrictions

13.1.5.1. Temperature Control Cluster (Server) Clarifications

Given that different markets have different customary methods of providing temperature settings (e.g. North America often prefers levels, whereas many other markets provide temperatures in °C), it is RECOMMENDED that when the Temperature Control cluster is present, the TemperatureLevel or TemperatureNumber feature of that cluster is chosen to follow the most widely applied convention for the market where the product is sold.

13.1.5.2. On/Off Cluster (Server) Clarifications

As indicated in the Element Requirements section below, the DF (Dead Front) feature is required for the On/Off cluster in this device type. See the "DeadFrontBehavior feature" section in the On/Off cluster description for detailed requirements. The "dead front" state is linked to the OnOff attribute in the On/Off cluster having the value False. Thus, the Off command of the On/Off cluster SHALL move the device into the "dead front" state, the On command of the On/Off cluster SHALL bring the device out of the "dead front" state, and the device SHALL adhere with the associated requirements on subscription handling and event reporting.

13.1.5.3. Best Effort Attribute Values in "Dead Front" State

When in "dead front", should the operational values of the cluster attributes not be available or accessible, the following are the RECOMMENDED best effort values for per cluster attributes when responding to a new subscription request or a read request. Attributes not listed have no change in their defined or expected values.

Cluster Name	Attribute	Fallback
Laundry Washer Mode	CurrentMode	MS
Laundry Washer Controls	SpinSpeedCurrent	null
	NumberOfRinses	null
	SpinSpeeds	null
	MaxRinses	null
Temperature Control	All attributes	MS
Identify	All attributes	MS
Operational State	PhaseList	null
	CurrentPhase	null
	CountdownTime	null
	OperationalState	Stopped
	OperationalError	No Error

13.1.6. Element Requirements

Below list qualities and conformance that override the cluster specification requirements. A blank table cell means there is no change to that item and the value from the cluster specification applies.

ID	Cluster	Element	Name	Constraint	Access	Conformance
0x0006	On/Off	Feature	DeadFrontBehavior			M
0x0051	Laundry Washer Mode	Attribute	StartUpMode			X
0x0051	Laundry Washer Mode	Feature	OnOff			X
0x0060	Operational State	Event	OperationCompletion			M

13.2. Refrigerator

A refrigerator represents a device that contains one or more cabinets that are capable of chilling or freezing food. Examples of consumer products that MAY make use of this device type include refrigerators, freezers, and wine coolers.

13.2.1. Refrigerator Architecture

A Refrigerator is always defined via endpoint composition. See the Section 13.2.5, “Device Type Requirements” section for more details.

A Refrigerator MAY include a semantic tag in the TagList attribute of the Descriptor cluster to describe the primary function of the device, e.g., "Refrigerator" or "Freezer".

An example of a Refrigerator with multiple cabinets is illustrated below.

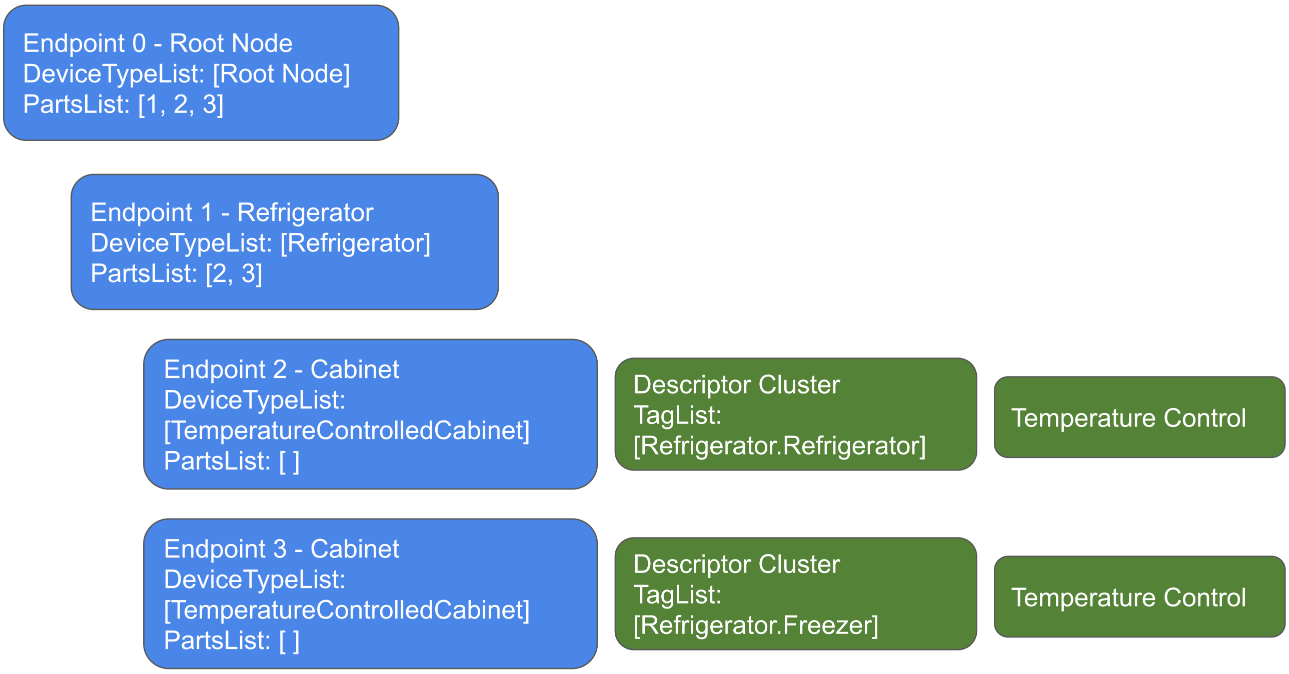

Endpoint 0 - Root Node
DeviceTypeList: [Root Node]
PartsList: [1, 2, 3]
Endpoint 1 - Refrigerator
DeviceTypeList: [Refrigerator]
PartsList: [2, 3]
Endpoint 2 - Cabinet
DeviceTypeList: [TemperatureControlledCabinet]
PartsList: [ ]
Descriptor Cluster
TagList: [Refrigerator.Refrigerator]
Temperature Control
Endpoint 3 - Cabinet
DeviceTypeList: [TemperatureControlledCabinet]
PartsList: [ ]
Descriptor Cluster
TagList: [Refrigerator.Freezer]
Temperature Control

Figure 6. Example of a Refrigerator

13.2.2. Revision History

This is the revision history for this document.

Revision	Description
1	Initial revision
2	Added Cooler requirement

13.2.3. Classification

ID	Device Name	Superset Of	Class	Scope
0x0070	Refrigerator		Simple	Endpoint

13.2.4. Conditions

A Refrigerator SHALL have the Cooler condition applied to at least one endpoint containing the Temperature Control Cluster.

See the Base Device Type definition for conformance tags.

13.2.5. Device Type Requirements

A Refrigerator SHALL be composed of at least one endpoint with the Temperature Controlled Cabinet device type as defined by the conformance below. There MAY be more endpoints with other device types existing in the Refrigerator.

If the Refrigerator contains more than one instance of a Temperature Controlled Cabinet, those instances SHALL include a semantic tag in the TagList attribute of the Descriptor cluster to disambiguate the cabinet, e.g., "freezer" or "refrigerator". Such a semantic tag SHALL be from either the defined Common or Refrigerator namespaces.

ID	Name	Constraint	Conformance
0x0071	Temperature Controlled Cabinet	min 1	M

13.2.6. Cluster Requirements

Each endpoint supporting the refrigerator device type MAY include these clusters based on the conformance defined below.

ID	Cluster	Client/Server	Quality	Conformance
0x0003	Identify	Server		O
0x0052	Refrigerator And Temperature Controlled Cabinet Mode	Server		O
0x0057	Refrigerator Alarm	Server		O

13.2.7. Element Requirements

Below list qualities and conformance that override the cluster specification requirements. A blank table cell means there is no change to that item and the value from the cluster specification applies.

ID	Cluster	Element	Name	Constraint	Access	Conformance
0x0052	Refrigerator And Temperature Controlled Cabinet Mode	Attribute	StartUpMode			X
0x0052	Refrigerator And Temperature Controlled Cabinet Mode	Feature	OnOff			X

13.3. Room Air Conditioner

This defines conformance to the Room Air Conditioner device type.

A Room Air Conditioner is a device with the primary function of controlling the air temperature in a single room.

13.3.1. Room Air Conditioner Architecture

A Room Air Conditioner is a device which at a minimum is capable of being turned on and off and of controlling the temperature in the living space.

A Room Air Conditioner MAY also support additional capabilities via endpoint composition. See the Section 13.3.5, "Device Type Requirements" section for typical device types.

The following diagram shows an example Room Air Conditioner consisting of a parent endpoint that is the Room Air Conditioner device type and several child endpoints providing additional capabilities. Note that two of the child endpoints are of the same device type, Temperature Sensor, which are being disambiguated via the requirements of endpoint composition defined in the system model.

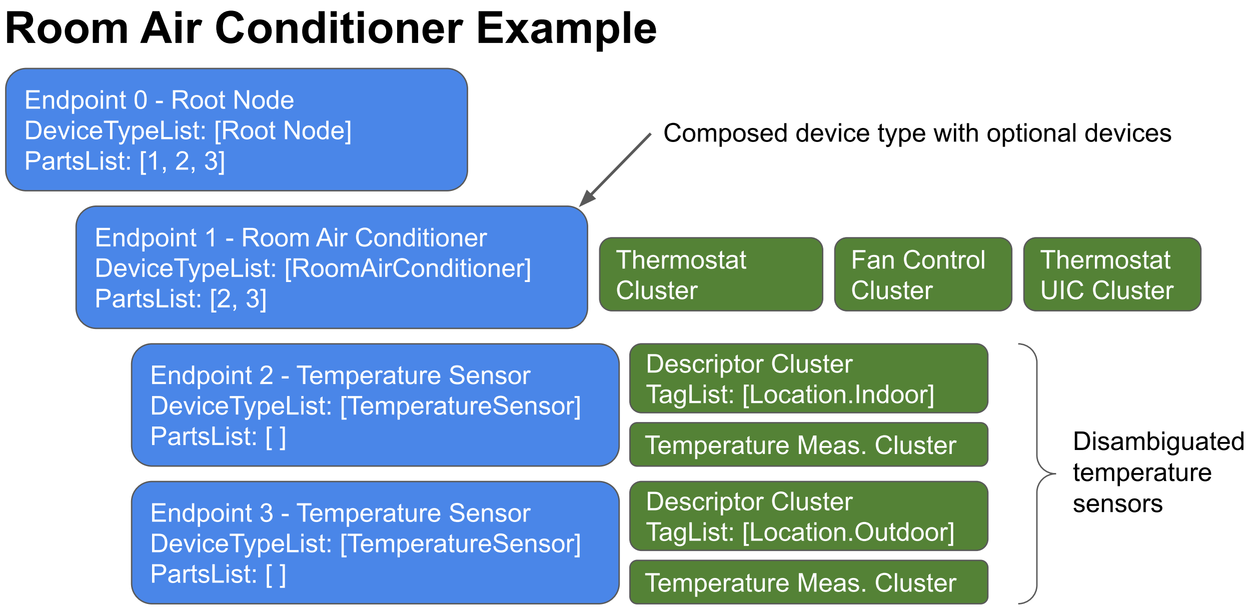

Room Air Conditioner Example
Endpoint 0 - Root Node
DeviceTypeList: [Root Node]
PartsList: [1, 2, 3]
Composed device type with optional devices
Endpoint 1 - Room Air Conditioner
DeviceTypeList: [RoomAirConditioner]
PartsList: [2, 3]
Thermostat Cluster
Fan Control Cluster
Thermostat UIC Cluster
Endpoint 2 - Temperature Sensor
DeviceTypeList: [TemperatureSensor]
PartsList: [ ]
Descriptor Cluster
TagList: [Location.Indoor]
Temperature Meas. Cluster
Endpoint 3 - Temperature Sensor
DeviceTypeList: [TemperatureSensor]
PartsList: [ ]
Descriptor Cluster
TagList: [Location.Outdoor]
Temperature Meas. Cluster
Disambiguated temperature sensors

Figure 7. Example of a Room Air Conditioner

13.3.2. Revision History

This is the revision history for this document.

Revision	Description
1	Initial revision
2	Thermostat User Interface Configuration cluster added; Updated the Scenes cluster to Scenes Management with Cluster ID: 0x0062

13.3.3. Classification

ID	Device Name	Superset Of	Class	Scope
0x0072	Room Air Conditioner		Simple	Endpoint

13.3.4. Conditions

See the Base Device Type definition for conformance tags.

13.3.5. Device Type Requirements

A Room Air Conditioner MAY have zero or more of each device type listed in this table subject to the conformance column of the table. All devices used in compositions SHALL adhere to the disambiguation requirements of the System Model. Additional device types not listed in this table MAY also be included in device compositions.

ID	Name	Constraint	Conformance
0x0302	Temperature Sensor		O
0x0307	Humidity Sensor		O

13.3.6. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

ID	Cluster	Client/Server	Conformance
0x0003	Identify	Server	M
0x0004	Groups	Server	O
0x0006	On/Off	Server	M
0x0062	Scenes Management	Server	O
0x0201	Thermostat	Server	M
0x0202	Fan Control	Server	O
0x0204	Thermostat User Interface Configuration	Server	O
0x0402	Temperature Measurement	Server	O
0x0405	Relative Humidity Measurement	Server	O

13.3.7. Cluster Restrictions

13.3.7.1. On/Off Cluster (Server) Clarifications

As indicated in the Element Requirements section below, the DF (Dead Front) feature is required for the On/Off cluster in this device type. See the "DeadFrontBehavior feature" section in the On/Off cluster description for detailed requirements. The "dead front" state is linked to the OnOff attribute in the On/Off cluster having the value False. Thus, the Off command of the On/Off cluster SHALL move the device into the "dead front" state, the On command of the On/Off cluster SHALL bring the device out of the "dead front" state, and the device SHALL adhere with the associated requirements on subscription handling and event reporting.

13.3.7.2. Best Effort Attribute Values in "Dead Front" State

When in "dead front", should the operational values of the cluster attributes not be available or accessible, the following are the RECOMMENDED best effort values for per cluster attributes when responding to a new subscription request or a read request. Attributes not listed have no change in their defined or expected values.

Cluster Name	Attribute	Fallback
Thermostat	LocalTemperature	null
Temperature Measurement	MeasuredValue	null
Relative Humidity Measurement	MeasuredValue	null
Fan Control	SpeedSetting	null
	PercentSetting	null

13.3.8. Element Requirements

This lists qualities and conformance that override the cluster specification requirements. A blank table cell means there is no change to that item and the value from the cluster specification applies.

ID	Cluster	Element	Name	Constraint	Access	Conformance
0x0006	On/Off	Feature	DeadFrontBehavior			M
0x0204	Thermostat User Interface Configuration	Attribute	KeypadLockout			O

13.4. Temperature Controlled Cabinet

A Temperature Controlled Cabinet only exists composed as part of another device type. It represents a single cabinet that is capable of having its temperature controlled. Such a cabinet may be chilling or freezing food, for example as part of a refrigerator, freezer, wine chiller, or other similar device. Equally, such a cabinet may be warming or heating food, for example as part of an oven, range, or similar device.

13.4.1. Revision History

This is the revision history for this document.

Revision	Description
1	Initial revision
2	Extension to heating cabinets
3	Added exclusivity for conditions
4	Mandate OperationCompletion Event for Oven Cavity Operational State cluster
5	Made TemperatureNumber (TN) the only valid temperature control mode

13.4.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x0071	Temperature Controlled Cabinet		Simple	Endpoint

13.4.3. Conditions

This device type SHALL support the following conformance conditions as defined below.

Condition	Description
Cooler	The device has cooling functionality.
Heater	The device has heating functionality.

Endpoints SHALL support at most one condition.

See the Base Device Type definition for conformance tags.

13.4.4. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

ID	Name	Client/Server	Quality	Conformance
0x0056	Temperature Control	Server		M
0x0402	Temperature Measurement	Server		O

ID	Name	Client/Server	Quality	Conformance
0x0052	Refrigerator And Temperature Controlled Cabinet Mode	Server		[Cooler]
0x0049	Oven Mode	Server		[Heater]
0x0048	Oven Cavity Operational State	Server		[Heater]

13.4.5. Element Requirements

Below list qualities and conformance that override the cluster specification requirements. A blank table cell means there is no change to that item and the value from the cluster specification applies.

ID	Cluster	Element	Name	Constraint	Access	Conformance
0x0052	Refrigerator And Temperature Controlled Cabinet Mode	Attribute	StartUpMode			X
0x0052	Refrigerator And Temperature Controlled Cabinet Mode	Feature	OnOff			X
0x0049	Oven Mode	Attribute	StartUpMode			X
0x0049	Oven Mode	Feature	OnOff			X
0x0048	Oven Cavity Operational State	Command	Pause			X
0x0048	Oven Cavity Operational State	Command	Resume			X
0x0048	Oven Cavity Operational State	Event	OperationCompletion			M
0x0056	Temperature Control	Feature	TemperatureNumber			M
0x0056	Temperature Control	Feature	TemperatureLevel			X

Temperature Controlled cabinets only allow the Temperature Control cluster to use the TemperatureNumber feature (i.e. actual temperature in °C). This is because using qualitative temperature levels (e.g. Low/Medium/High) does not allow the behavior expected by the majority of clients. Clients would be trying to "set the temperature" of a cabinet using that cluster, such as an oven's cooking temperature, or a refrigerator's internal cabinet temperature setpoint.

13.5. Dishwasher

A dishwasher is a device that is generally installed in residential homes and is capable of washing dishes, cutlery, and other items associate with food preparation and consumption. The device can be permanently installed or portable and can have variety of filling and draining methods.

13.5.1. Revision History

This is the revision history for this document.

Revision	Description
1	Initial revision
2	Mandate OperationCompletion Event

13.5.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x0075	Dishwasher		Simple	Endpoint

13.5.3. Conditions

See the Base Device Type definition for conformance tags.

13.5.4. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

ID	Cluster	Client/Server	Conformance
0x0003	Identify	Server	O
0x0006	On/Off	Server	O
0x0056	Temperature Control	Server	O
0x0059	Dishwasher Mode	Server	O
0x005D	Dishwasher Alarm	Server	O
0x0060	Operational State	Server	M

Notes

- A dishwasher cycle is a combination of a mode (if supported) and a temperature (if supported).

The operational state cluster is then used to start the cycle once these selections have been made via the client.

13.5.5. Cluster Restrictions

13.5.5.1. Temperature Control Cluster (Server) Clarifications

Given that different markets have different customary methods of providing temperature settings (e.g. North America often prefers levels, whereas many other markets provide temperatures in °C), it is RECOMMENDED that when the Temperature Control cluster is present, the TemperatureLevel or TemperatureNumber feature of that cluster is chosen to follow the most widely applied convention for the market where the product is sold.

13.5.5.2. On/Off Cluster (Server) Clarifications

As indicated in the Element Requirements section below, the DF (Dead Front) feature is required for the On/Off cluster in this device type. See the "DeadFrontBehavior feature" section in the On/Off cluster description for detailed requirements. The "dead front" state is linked to the OnOff attribute in the On/Off cluster having the value False. Thus, the Off command of the On/Off cluster SHALL move the device into the "dead front" state, the On command of the On/Off cluster SHALL bring the device out of the "dead front" state, and the device SHALL adhere with the associated requirements on subscription handling and event reporting.

13.5.5.3. Best Effort Attribute Values in "Dead Front" State

When in "dead front", should the operational values of the cluster attributes not be available or accessible, the following are the RECOMMENDED best effort values for per cluster attributes when responding to a new subscription request or a read request. Attributes not listed have no change in their defined or expected values.

Cluster Name	Attribute	Fallback
Dishwasher Mode	CurrentMode	MS
Temperature Control	All attributes	MS
Dishwasher Operational State	PhaseList	null
	CurrentPhase	null
	CountdownTime	null
	OperationalState	Stopped
	OperationalError	No Error

13.5.6. Element Requirements

This lists qualities and conformance that override the cluster specification requirements. A blank table cell means there is no change to that item and the value from the cluster specification applies.

ID	Cluster	Element	Name	Constraint	Access	Conformance
0x0006	On/Off	Feature	DeadFront-Behavior			M
0x0059	Dishwasher Mode	Attribute	StartUpMode			X
0x0059	Dishwasher Mode	Feature	OnOff			X
0x0060	Operational State	Event	OperationCompletion			M

13.6. Laundry Dryer

A Laundry Dryer represents a device that is capable of drying laundry items.

13.6.1. Revision History

This is the revision history for this document.

Revision	Description
1	Initial revision
2	Mandate OperationCompletion Event

13.6.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x007C	Laundry Dryer		Simple	Endpoint

13.6.3. Conditions

See the Base Device Type definition for conformance tags.

13.6.4. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

ID	Name	Client/Server	Quality	Conformance
0x0003	Identify	Server		O
0x0051	Laundry Washer Mode	Server		O
0x0006	On/Off	Server		O

ID	Name	Client/Server	Quality	Conformance
0x004A	Laundry Dryer Controls	Server		O
0x0056	Temperature Control	Server		O
0x0060	Operational State	Server		M

13.6.5. Cluster Restrictions

13.6.5.1. Temperature Control Cluster (Server) Clarifications

Given that different markets have different customary methods of providing temperature settings (e.g. North America often prefers levels, whereas many other markets provide temperatures in °C), it is RECOMMENDED that when the Temperature Control cluster is present, the TemperatureLevel or TemperatureNumber feature of that cluster is chosen to follow the most widely applied convention for the market where the product is sold.

13.6.5.2. On/Off Cluster (Server) Clarifications

The actions carried out by a Laundry Dryer device on receipt of specific commands are shown below. As indicated in the Element Requirements section below, the DF (Dead Front) feature is required for the On/Off cluster in this device type. See the "DeadFrontBehavior feature" section in the On/Off cluster description for detailed requirements. The "dead front" state is linked to the OnOff attribute in the On/Off cluster having the value False. Thus, the Off command of the On/Off cluster SHALL move the device into the "dead front" state, the On command of the On/Off cluster SHALL bring the device out of the "dead front" state, and the device SHALL adhere with the associated requirements on subscription handling and event reporting.

13.6.5.3. Best Effort Attribute Values in "Dead Front" State

When in "dead front", should the operational values of the cluster attributes not be available or accessible, the following are the RECOMMENDED best effort values for per cluster attributes when responding to a new subscription request or a read request. Note that some of these attributes may be missing for the clusters not implemented on the endpoint due to optionality.

Cluster Name	Attribute	Value
Laundry Dryer Mode	CurrentMode	Manufacturer Specific
Laundry Dryer Controls	TemperatureLevel	Null
	DrynessLevel	Null
Temperature Control	All attributes	Manufacturer Specific

Cluster Name	Attribute	Value
Operational State	PhaseList	Null
	CurrentPhase	Null
	CountdownTime	Null
	OperationalStateList	Fully populated
	OperationalState	Stopped
	OperationalError	No Error

13.6.6. Element Requirements

Below list qualities and conformance that override the cluster specification requirements. A blank table cell means there is no change to that item and the value from the cluster specification applies.

ID	Cluster	Element	Name	Constraint	Access	Conformance
0x0006	On/Off	Feature	DeadFrontBehavior			M
0x0051	Laundry Washer Mode	Attribute	StartUpMode			X
0x0051	Laundry Washer Mode	Feature	OnOff			X
0x0060	Operational State	Event	OperationCompletion			M

13.7. Cook Surface

A Cook Surface device type represents a heating object on a cooktop or other similar device. It SHALL only be used when composed as part of another device type.

13.7.1. Revision History

This is the revision history for this document.

Revision	Description
1	Initial revision
2	Made TemperatureLevel (TL) the only valid temperature control mode

13.7.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x0077	Cook Surface		Simple	Endpoint

13.7.3. Conditions

See the Base Device Type definition for conformance tags.

13.7.4. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

ID	Name	Client/Server	Quality	Conformance
0x0056	Temperature Control	Server		O.a+
0x0402	Temperature Measurement	Server		O.a+
0x0006	On/Off	Server		O

13.7.5. Cluster Restrictions

13.7.5.1. On/Off Cluster (Server) Clarifications

The OffOnly feature is required for the On/Off cluster in this device type due to safety requirements.

13.7.6. Element Requirements

Below list qualities and conformance that override the cluster specification requirements. A blank table cell means there is no change to that item and the value from the cluster specification applies.

ID	Cluster	Element	Name	Constraint	Access	Conformance
0x0006	On/Off	Feature	OffOnly			M
0x0056	Temperature Control	Feature	TemperatureLevel			M
0x0056	Temperature Control	Feature	TemperatureNumber			X

Whenever the Temperature Control cluster is included on a Cook Surface, the Temperature Control cluster SHALL use the TemperatureLevel feature rather than the TemperatureNumber feature. This is because users are usually in the loop for controlling the temperature of the food being cooked within a heated cooking utensil. For example, while the surface temperature of a cooktop may be significantly above 100°C, an open pot of water will never exceed the boiling point of water as all excess energy transmitted is spent on the water’s phase change to steam and the liquid within the pot reaches an equilibrium temperature.

13.8. Cooktop

A cooktop is a cooking surface that heats food either by transferring currents from an electromagnetic field located below the glass surface directly to the magnetic induction cookware placed above or through traditional gas or electric burners.

13.8.1. Revision History

This is the revision history for this document.

Revision	Description
1	Initial revision

13.8.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x0078	Cooktop		Simple	Endpoint

13.8.3. Conditions

See the Base Device Type definition for conformance tags.

13.8.4. Device Type Requirements

A Cooktop SHALL be composed of zero or more endpoints with the Cook Surface device type as defined by the conformance below.

A cooktop falls under strict regulatory control in some regions. One of these restrictions for non-induction cooktops is that the only remote commands available are to turn off the entire device or read out the temperature setting. This one scenario for allowed remote operation is specifically to address the use case of a device that is left on after a user leaves the home. The individual cooking surfaces cannot be shut off. This leads to a model of a cooktop that has 0 controllable cooking surfaces. For example, the requirements that exist for a gas cooktop would result in zero Cook Surface instances being exposed.

If the Cooktop contains more than one instance of a Cook Surface, those instances SHALL include a semantic tag in the TagList attribute of the Descriptor cluster to disambiguate the cook surface, e.g., "front", "left", or "back". Such a semantic tag SHALL be from the Common namespaces.

ID	Name	Constraint	Conformance
0x0077	Cook Surface	min 1	O

13.8.5. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

ID	Name	Client/Server	Quality	Conformance
0x0003	Identify	Server		O
0x0006	On/Off	Server		M

13.8.6. Cluster Restrictions

13.8.6.1. On/Off Cluster (Server) Clarifications

The OffOnly feature is required for the On/Off cluster in this device type due to safety requirements.

13.8.7. Element Requirements

Below list qualities and conformance that override the cluster specification requirements. A blank table cell means there is no change to that item and the value from the cluster specification applies.

ID	Cluster	Element	Name	Constraint	Access	Conformance
0x0006	On/Off	Feature	OffOnly			M

13.9. Oven

An oven represents a device that contains one or more cabinets, and optionally a single cooktop, that are all capable of heating food. Examples of consumer products implementing this device type include ovens, wall ovens, convection ovens, etc.

13.9.1. Oven Architecture

An oven is always defined via endpoint composition. See Section 13.9.5, “Device Type Requirements” for more details.

An example of an oven with two cabinets (one above the other) and a cooktop (with two cook surfaces) is illustrated below.

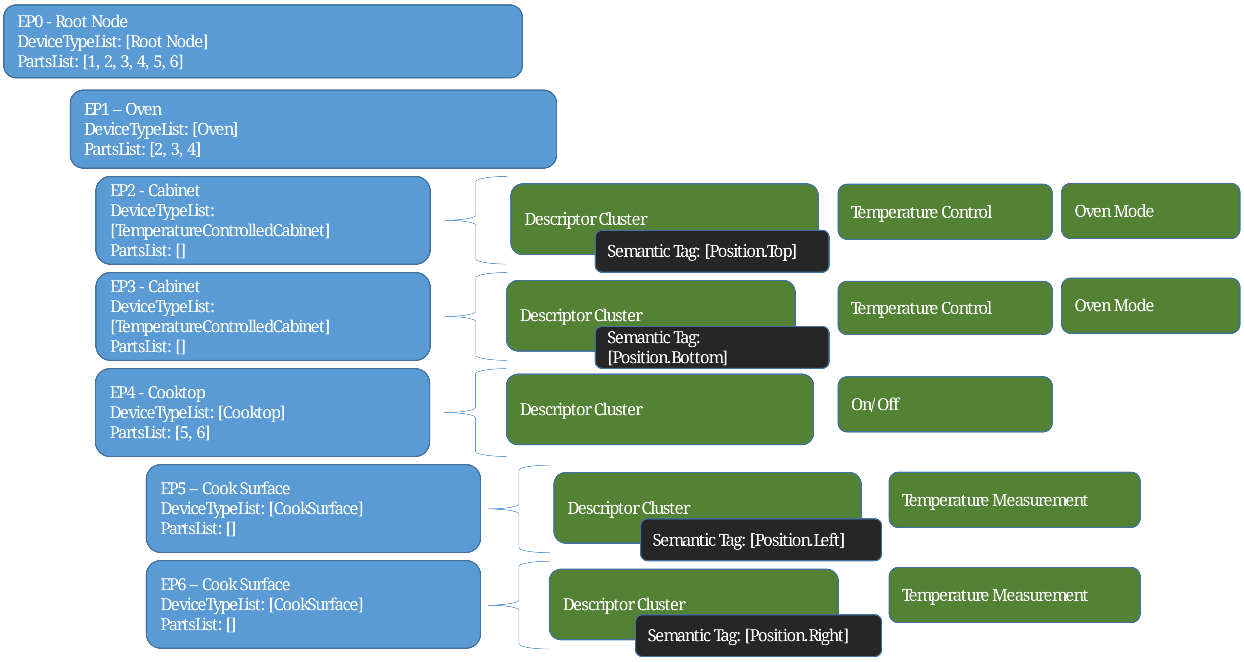


Figure 8. Example of an Oven

13.9.2. Revision History

This is the revision history for this document.

Revision	Description
1	Initial revision
2	Added Heater requirement

13.9.3. Classification

ID	Device Name	Superset Of	Class	Scope
0x007B	Oven		Simple	Endpoint

13.9.4. Conditions

An Oven SHALL have the Heater condition applied to at least one endpoint containing the Temperature Control Cluster.

See the Base Device Type definition for conformance tags.

13.9.5. Device Type Requirements

An Oven SHALL be composed of at least one endpoint with Temperature Controlled Cabinet device type. There MAY be more endpoints with other device types existing in the Oven. Note that any instance of the TemperatureControl cluster on an endpoint is scoped to the device type on that endpoint, and not the whole node.

If the Oven contains more than one instance of a Temperature Controlled Cabinet, those instances SHALL include a semantic tag in the TagList attribute of the Descriptor cluster to disambiguate the cabinet, e.g., "Top" or "Bottom". Such a semantic tag SHALL be from the defined Common Position namespaces.

Regional restrictions and safety regulations may dictate which aspects of a Temperature Controlled Cabinet may be remotely accessible. In such cases, clusters exposed by an instance of a Temperature Controlled Cabinet MAY have limitations on what commands are supported or what attributes are mutable.

ID	Name	Constraint	Conformance
0x0071	Temperature Controlled Cabinet	min 1	M
0x0078	Cooktop	max 1	O

13.9.6. Cluster Requirements

Each endpoint supporting this device type MAY include these clusters based on the conformance defined below.

ID	Name	Client/Server	Quality	Conformance
0x0003	Identify	Server		O

13.10. Extractor Hood

An Extractor Hood is a device that is generally installed above a cooking surface in residential kitchens. An Extractor Hood’s primary purpose is to reduce odors that arise during the cooking process by either extracting the air above the cooking surface or by recirculating and filtering it. It may also contain a light for illuminating the cooking surface.

Extractor Hoods may also be known by the following names:

- Hoods
- Extractor Fans
- Extractors
- Range Hoods
- Telescoping Hoods
- Telescoping Extractors

13.10.1. Revision History

This is the revision history for this document.

Revision	Description
1	Initial revision

13.10.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x007A	Extractor Hood		Simple	Endpoint

13.10.3. Conditions

See the Base Device Type definition for conformance tags.

13.10.4. Device Type Requirements

An Extractor Hood is composed of other device types listed in this table subject to the conformance column of the table. All devices used in compositions SHALL adhere to the disambiguation and superset requirements of the System Model. Specifically, please note that the On/Off Light as listed is a Superset Device Type as defined by the System Model (see *Superset Device Types* in [Matter-Core]), and so the rules defined in that section apply to the use of On/Off Light as a superset when composed in this device type. Additional device types not listed in this table MAY also be included in device compositions.

ID	Name	Constraint	Conformance
0x0100+	On/Off Light+		O

13.10.5. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

ID	Name	Client/Server	Conformance
0x0003	Identify	Server	O
0x0071	HEPA Filter Monitoring	Server	O
0x0072	Activated Carbon Filter Monitoring	Server	O
0x0202	Fan Control	Server	M

13.10.6. Element Requirements

This lists qualities and conformance that override the cluster specification requirements. A blank table cell means there is no change to that item and the value from the cluster specification applies.

ID	Cluster	Element	Name	Constraint	Access	Conformance
0x0202	Fan Control	Feature	Rocking			X
0x0202	Fan Control	Feature	Wind			X
0x0202	Fan Control	Feature	AirflowDirection			X

13.11. Microwave Oven

This defines conformance to the Microwave Oven device type.

A Microwave Oven is a device with the primary function of heating foods and beverages using a magnetron.

13.11.1. Microwave Oven Architecture

A Microwave Oven is a device which at a minimum is capable of being started and stopped and of setting a power level.

A Microwave Oven MAY also support additional capabilities via endpoint composition. See the Device Type Requirements section for typical device types.

The following diagram shows an example Microwave Oven consisting of a parent endpoint that is the Microwave Oven device type and a child endpoint providing additional capabilities.

A microwave oven placed above a thermal oven or cooktop/hob may also include a light for illuminating the cooking surface of the thermal oven or cooktop/hob and an exhaust fan for removing cooking odors.

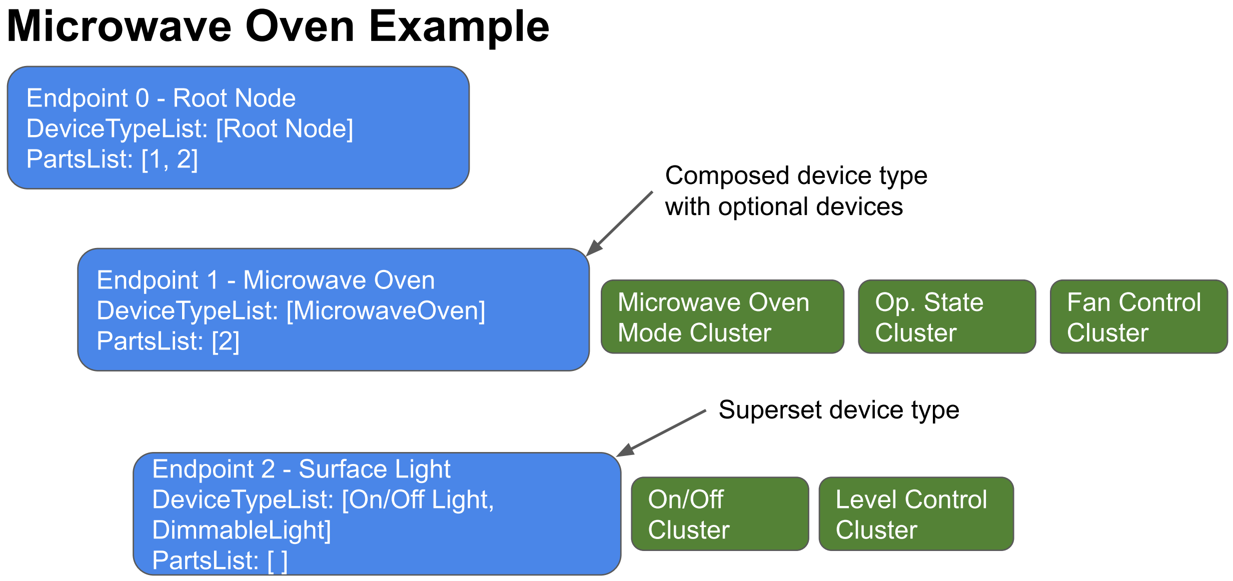


Figure 9. Example of a Microwave Oven

13.11.2. Revision History

This is the revision history for this document.

Revision	Description
1	Initial revision
2	Mandate OperationCompletion Event

13.11.3. Classification

ID	Device Name	Superset Of	Class	Scope
0x0079	Microwave Oven		Simple	Endpoint

13.11.4. Conditions

See the Base Device Type definition for conformance tags.

13.11.5. Device Type Requirements

Each endpoint supporting this device type SHALL include endpoints with these device types based on the conformance defined below.

ID	Name	Constraint	Conformance
0x0100+	On/Off Light+		O

When a light is included as part of a composed device type, it is intended to be used as surface light when the microwave oven is installed above a range in an "over the range" configuration rather than the internal light of the microwave oven cavity.

13.11.6. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

ID	Name	Client/Server	Conformance
0x0003	Identify	Server	O
0x0060	Operational State	Server	M
0x0202	Fan Control	Server	O
0x005E	Microwave Oven Mode	Server	M
0x005F	Microwave Oven Control	Server	M

When the Fan Control cluster is supported on an endpoint of this device type, it is intended to be used as a ventilation fan when the microwave oven is installed above a range in an "over the range" configuration rather than the internal fan of the microwave oven cavity.

13.11.7. Element Requirements

This lists qualities and conformance that override the cluster specification requirements. A blank table cell means there is no change to that item and the value from the cluster specification applies.

ID	Cluster	Element	Name	Constraint	Access	Conformance
0x0060	Operational State	Attribute	CountdownTime			M
0x0060	Operational State	Event	OperationCompletion			M
0x0202	Fan Control	Feature	Wind			X
0x0202	Fan Control	Feature	AirflowDirection			X

13.11.8. Cluster Usage

This section describes how to control and monitor the operation of a Microwave Oven device. This information is meant to clarify how the data dependencies within the device type’s cluster composition are to be used.

Note that the device operations may also be the result of, or affected by, out-of-band actions such as physical button presses on the device, internally scheduled events, vendor application requests, commands invoked via other fabrics, internal device timeouts, etc. For example, a user may pause the oven during operation by opening the door to check on the food.

13.11.8.1. Starting the Oven

The oven operational attributes are set by sending the SetCookingParameters command of the Microwave Oven Control cluster with the values as intended by the user via a client. Oven operation can be started by sending the Start command via the Operational State cluster or one of its derivatives, if supported, or within the SetCookingParameters command via the StartAfterSetting attribute, if supported.

Upon setting the CookTime attribute via the SetCookingParameters command, the CountdownTime attribute of the Operational State cluster or one of its derivatives , if supported, is set to the same value as the CookTime attribute.

Once oven operation is started, the values previously sent by the SetCookingParameters command are used to control the oven operation and the CountdownTime attribute of the Operational State cluster or one of its derivatives begins counting down.

13.11.8.2. During Operation

While the oven is in the Running state, the CountdownTime attribute of the Operational State cluster or one of its derivatives counts down and the CookTime attribute of the Microwave Oven Control cluster remains fixed.

13.11.8.3. Stopping the Oven

Oven operation will end when either the Stop command of the Operational State cluster or one of its derivatives, if supported, is sent, the CountdownTime value reaches zero, or the oven is stopped via an out-of-band method.

It is recommended that when the oven enters the Stopped state of the Operational State cluster or one of its derived clusters, the attribute values of the Microwave Oven Control cluster be set to their default values by the server.

13.11.8.4. Adding More Time

When time is added to the CookTime attribute using the AddMoreTime command of the Microwave Oven Control cluster, the same amount of time is also added to the CountdownTime attribute of the Operational State cluster or one of its derivatives. See the CookTime attribute constraints and AddMoreTime command for more details.

Chapter 14. Energy Device Types

14.1. EVSE Device Type

An EVSE (Electric Vehicle Supply Equipment) is a device that allows an EV (Electric Vehicle) to be connected to the mains electricity supply to allow it to be charged (or discharged in case of Vehicle to Grid / Vehicle to Home applications).

14.1.1. EVSE Architecture

An EVSE is always defined via endpoint composition. See the Section 14.1.5, “Device Type Requirements” section for more details.

An example of an EVSE with single phase AC supply is illustrated below.

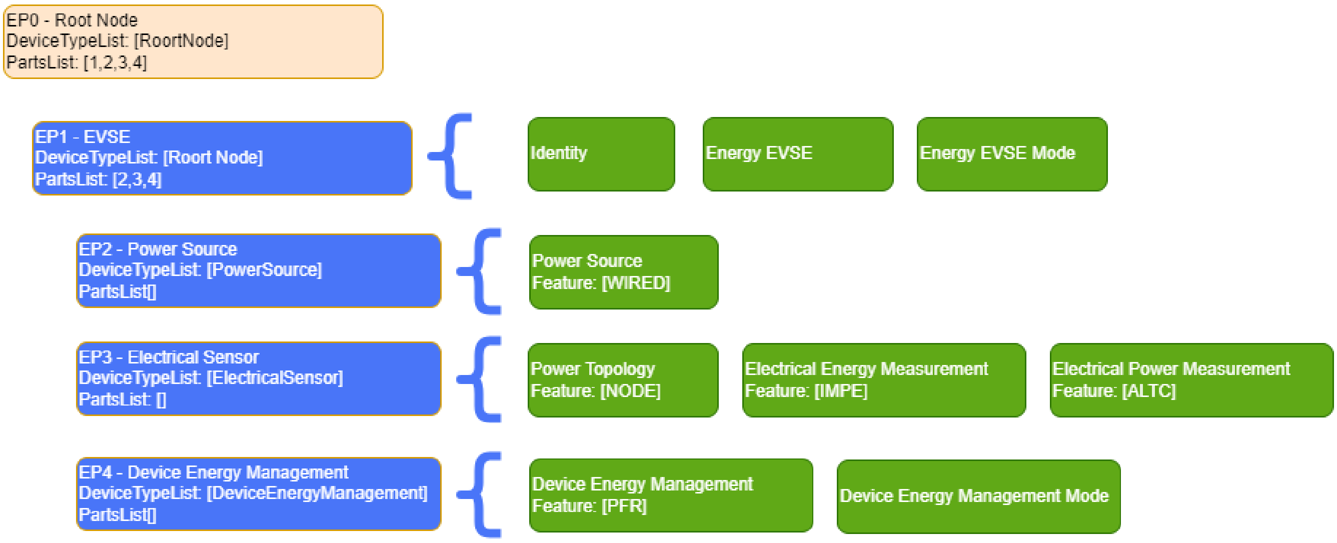


Figure 10. Example of a single phase EVSE

The EVSE may also indicate its internal temperature using the temperature measurement cluster (not shown).

An example of an EVSE with a 3 phase AC supply is illustrated below.

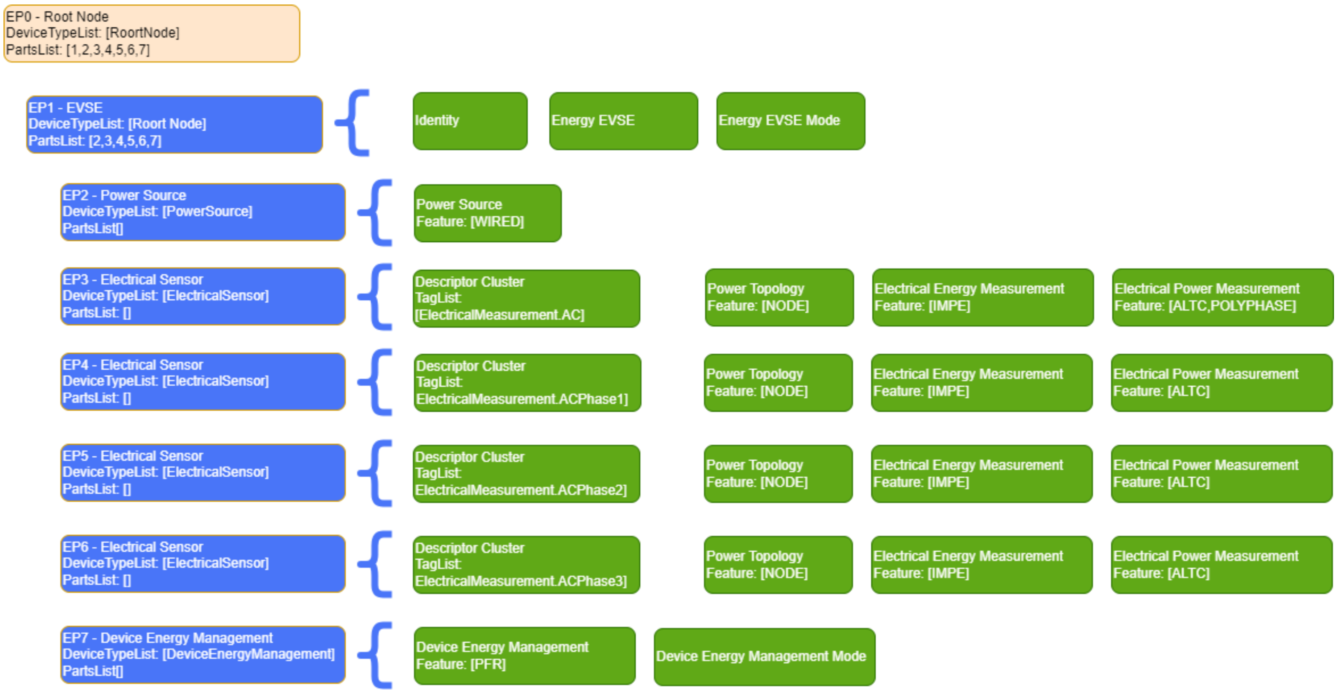

EP0 - Root Node
DeviceTypeList: [RoortNode]
PartsList: [1,2,3,4,5,6,7]
EP1 - EVSE
DeviceTypeList: [Roort Node]
PartsList: [2,3,4,5,6,7]
Identity
Energy EVSE
Energy EVSE Mode
EP2 - Power Source
DeviceTypeList: [PowerSource]
PartsList[]
Power Source
Feature: [WIRED]
EP3 - Electrical Sensor
DeviceTypeList: [ElectricalSensor]
PartsList: []
Descriptor Cluster
TagList:
[ElectricalMeasurement.AC]
Power Topology
Feature: [NODE]
Electrical Energy Measurement
Feature: [IMPE]
Electrical Power Measurement
Feature: [ALTC,POLYPHASE]
EP4 - Electrical Sensor
DeviceTypeList: [ElectricalSensor]
PartsList: []
Descriptor Cluster
TagList:
ElectricalMeasurement.ACPhase1]
Power Topology
Feature: [NODE]
Electrical Energy Measurement
Feature: [IMPE]
Electrical Power Measurement
Feature: [ALTC]
EP5 - Electrical Sensor
DeviceTypeList: [ElectricalSensor]
PartsList: []
Descriptor Cluster
TagList:
ElectricalMeasurement.ACPhase2]
Power Topology
Feature: [NODE]
Electrical Energy Measurement
Feature: [IMPE]
Electrical Power Measurement
Feature: [ALTC]
EP6 - Electrical Sensor
DeviceTypeList: [ElectricalSensor]
PartsList: []
Descriptor Cluster
TagList:
ElectricalMeasurement.ACPhase3]
Power Topology
Feature: [NODE]
Electrical Energy Measurement
Feature: [IMPE]
Electrical Power Measurement
Feature: [ALTC]
EP7 - Device Energy Management
DeviceTypeList: [DeviceEnergyManagement]
PartsList[]
Device Energy Management
Feature: [PFR]
Device Energy Management Mode

Figure 11. Example of a 3 phase EVSE

14.1.2. Revision History

This is the revision history for this document.

Revision	Description
1	Initial revision
2	Addition of associated Device Energy Management device

14.1.3. Classification

ID	Device Name	Superset Of	Class	Scope
0x050C	Energy EVSE		Simple	Endpoint

14.1.4. Conditions

See the Base Device Type definition for conformance tags.

14.1.5. Device Type Requirements

An EVSE SHALL be composed of at least one endpoint with device types as defined by the conformance below. There MAY be more endpoints with other device types existing in the EVSE.

ID	Name	Constraint	Conformance
0x0011	Power Source	min 1	M
0x0510	Electrical Sensor	min 1	M

ID	Name	Constraint	Conformance
0x050D	Device Energy Management	min 1	M

14.1.5.1. Cluster Requirements on Composing Device Types

The Electrical Sensor device SHALL include both the Electrical Energy Measurement and Electrical Power Measurement clusters, measuring the total energy and power of the EVSE.

If an EVSE supports three phase power then it SHALL include three additional endpoints including an Electrical Sensor Device Type as child elements. For each child endpoint it SHALL include a semantic tag from the Electrical Measurement Namespace in the TagList attribute of the Descriptor cluster to describe the endpoint for the relevant Electrical Power Measurement and Electrical Energy Measurement clusters indicating the relevant AC phase that is being measured.

The Device Energy Management cluster included in the Device Energy Management device SHALL support the Power Forecast Reporting (PFR) feature. If the EVSE supports the V2X feature then the Device Energy Management cluster included in the Device Energy Management device SHALL support the Power Adjustment (PA) feature.

14.1.6. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

ID	Name	Client/Server	Conformance
0x0003	Identify	Server	O
0x0099	Energy EVSE	Server	M
0x009D	Energy EVSE Mode	Server	M
0x0402	Temperature Measurement	Server	O

14.2. Water Heater Device Type

A water heater is a device that is generally installed in properties to heat water for showers, baths etc.

14.2.1. Water Heater Architecture

A Water Heater is always defined via endpoint composition.

If a Water Heater supports multiple temperature measurement sensors as child elements, it SHALL include a separate endpoint for each sensor. Each endpoint SHALL include a semantic tag in the TagList attribute of the Descriptor cluster to describe the relevant position of the sensor. Such a semantic tag SHALL be from the defined Common Position namespace (i.e. Top, Middle, Bottom etc).

For basic control features the Water Heater re-uses the Thermostat cluster with the **HEAT** and **SCH** features. This allows it to have daily schedules set as to when the hot water heating is enabled, as well as setting the desired setpoint of the hot water.

Additional features of the Water Heater (such as reporting estimated hot water content, and smart reheating functions) are provided by the Water Heater application cluster.

In order to add energy management capability, the Device Energy Management cluster may be optionally supported, and if so, the Electrical Power Measurement and Electrical Energy Measurement clusters are supported via the Electrical Sensor device type.

An example of a Water Heater device is illustrated below.

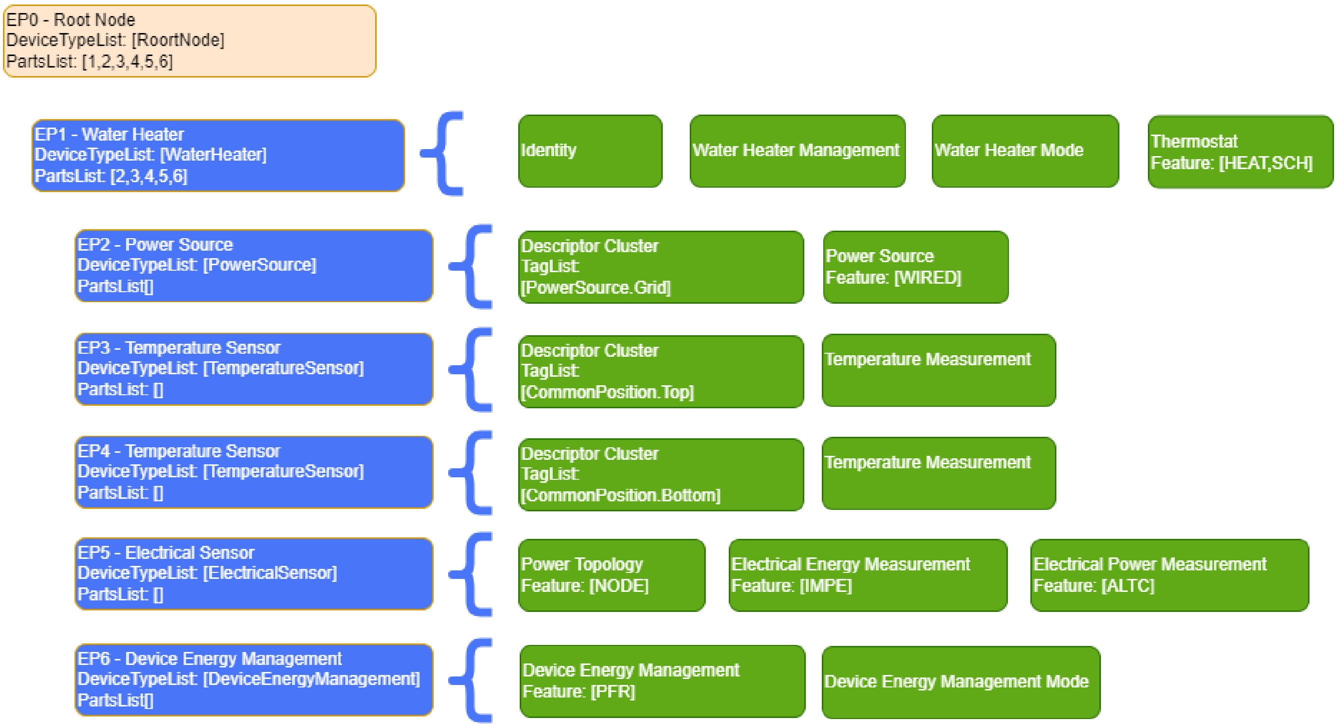


Figure 12. Example of a Water Heater

14.2.2. Revision History

This is the revision history for this document.

Revision	Description
1	Initial revision

14.2.3. Classification

ID	Device Name	Superset Of	Class	Scope
0x050F	Water Heater		Simple	Endpoint

14.2.4. Conditions

See the Base Device Type definition for conformance tags.

14.2.5. Device Type Requirements

A Water Heater SHALL be composed of at least one endpoint with device types as defined by the conformance below. There MAY be more endpoints with other device types existing in the Water Heater.

ID	Name	Constraint	Conformance
0x0011	Power Source		O
0x0302	Temperature Sensor		O
0x0510	Electrical Sensor		desc
0x050D	Device Energy Management		O

14.2.5.1. Electrical Sensor Device Type

If a Device Energy Management device type is included as part of a composition, the Electrical Sensor device type SHALL also be included.

14.2.5.2. Cluster Requirements on Composing Device Types

If an Electrical Sensor device is included as part of a composition, it SHALL include both the Electrical Energy Measurement and Electrical Power Measurement clusters, measuring the total energy and power of the Water Heater.

If a Device Energy Management device type is included on a separate endpoint as part of a composition and the Device Energy Management cluster is supported on the same endpoint, the Power Forecast Reporting feature of the Device Energy Management cluster SHALL also be supported.

14.2.6. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

ID	Name	Client/Server	Conformance
0x0003	Identify	Server	O
0x0201	Thermostat	Server	M
0x0094	Water Heater Management	Server	M
0x009E	Water Heater Mode	Server	M

14.2.7. Element Requirements

This lists qualities and conformance that override the cluster specification requirements. A blank table cell means there is no change to that item and the value from the cluster specification applies.

ID	Cluster	Element	Name	Constraint	Access	Conformance
0x0201	Thermostat	Feature	Heating			M

The Energy Management feature of the Water Heater cluster SHALL be supported if the Device Energy Management device type is included.

If Off is a supported SystemMode in the Thermostat cluster, setting the SystemMode of the Thermostat cluster to Off SHALL set the CurrentMode attribute of the Water Heater Mode cluster to a mode having the Off mode tag value and vice versa.

At least one entry in the SupportedModes attribute of the Water Heater Mode cluster SHALL include the Timed mode tag in the ModeTags field list.

14.3. Solar Power Device Type

A Solar Power device is a device that allows a solar panel array, which can optionally be comprised of a set parallel strings of solar panels, and its associated controller and, if appropriate, inverter, to be monitored and controlled by an Energy Management System.

14.3.1. Solar Power Architecture

A Solar Power device is always defined via endpoint composition. See the Section 14.3.5, “Device Type Requirements” section for more details.

An example of a Solar Power device with single phase AC output is illustrated below.

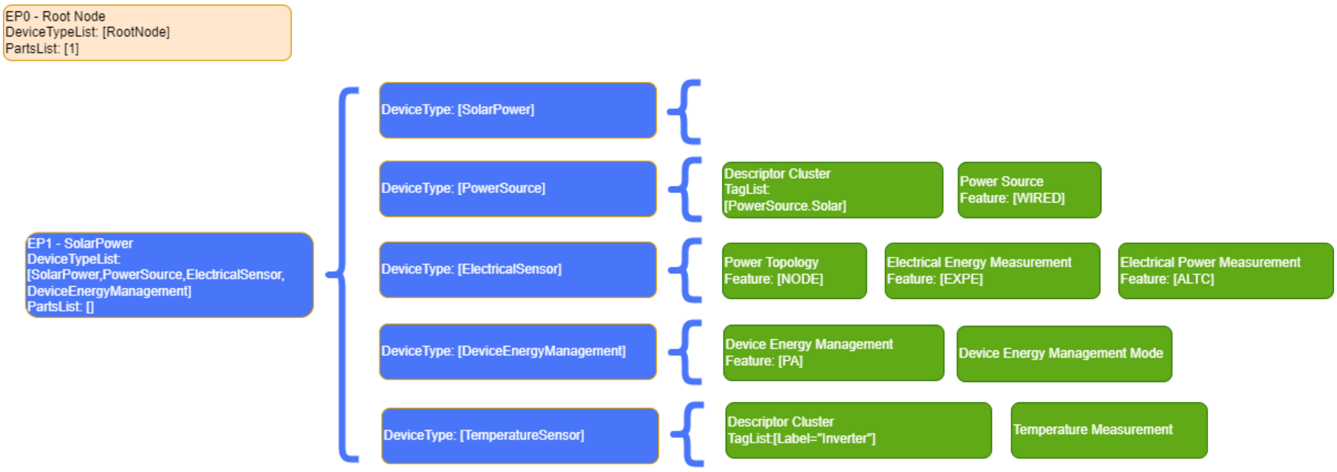


Figure 13. Example of a Solar Power device with single phase AC output

An example of a Solar Power device with single phase AC output, but with the ability to measure the output from 4 sets of solar panels supplying the overall device is illustrated below.

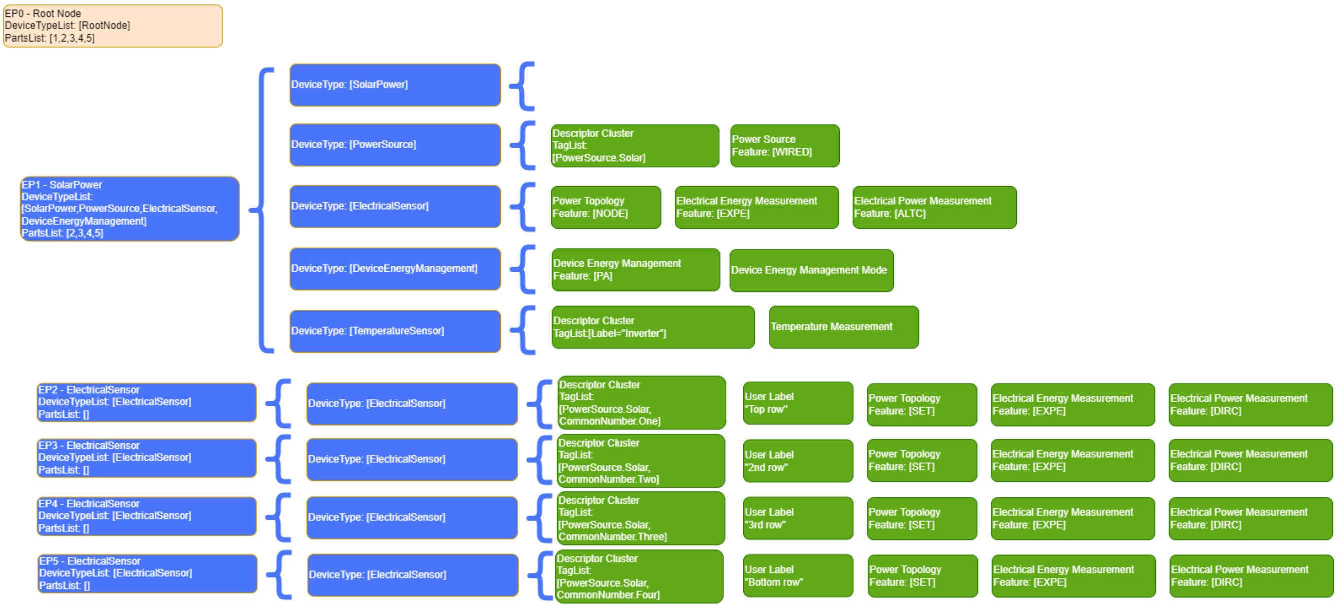


Figure 14. Example of a Solar Power device with solar panel measurements

An example of a Solar Power device with single phase AC output, but with the ability to measure the output from 4 individual solar panels, arranged as 2 strings or 2 panels each is illustrated below.

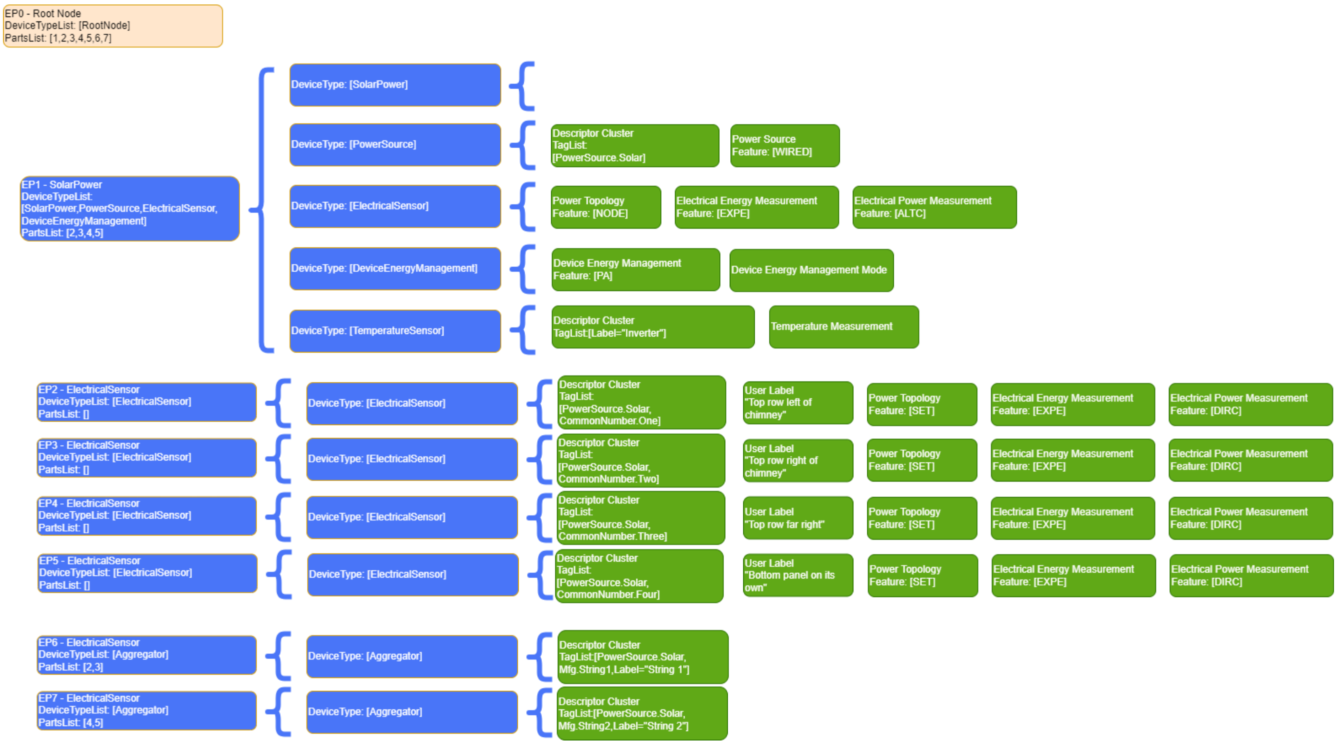


Figure 15. Example of a Solar Power device with solar panel measurements

14.3.2. Revision History

This is the revision history for this document.

Revision	Description
1	Initial revision

14.3.3. Classification

ID	Device Name	Superset Of	Class	Scope
0x0017	Solar Power		Simple	Endpoint

14.3.4. Conditions

See the Base Device Type definition for conformance tags.

14.3.5. Device Type Requirements

A Solar Power device SHALL be composed of at least one endpoint with device types as defined by the conformance below. There MAY be more endpoints with additional instances of these device types or additional device types existing in the Solar Power device.

ID	Name	Constraint	Conformance
0x0011	Power Source	min 1	M
0x0510	Electrical Sensor	min 1	M
0x050D	Device Energy Management		O
0x0302	Temperature Sensor		O

14.3.5.1. Cluster Requirements on Composing Device Types

Below list qualities and conformance that override the cluster specification requirements for the composing device types. A blank table cell means there is no change to that item and the value from the cluster specification applies.

Device ID	Device	Cluster ID	Cluster	Element	Name	Constraint	Access	Conformance
0x0011	Power Source	0x002F	Power Source	Feature	Wired			M
0x0011	Power Source	0x001D	Descriptor	Feature	TagList			M
0x0510	Electrical Sensor	0x0090	Electrical Power Measurement					M
0x0510	Electrical Sensor	0x0090	Electrical Power Measurement	Attribute	Voltage			M

Device ID	Device	Cluster ID	Cluster	Element	Name	Constraint	Access	Conformance
0x0510	Electrical Sensor	0x0090	Electrical Power Measurement	Attribute	ActiveCurrent			M
0x0510	Electrical Sensor	0x0091	Electrical Energy Measurement					M
0x0510	Electrical Sensor	0x0091	Electrical Energy Measurement	Feature	ExportedEnergy			M
0x050D	Device Energy Management	0x0098	Device Energy Management	Feature	PowerAdjustment			M
0x0302	Temperature Sensor	0x001D	Descriptor	Feature	TagList			M

The Descriptor cluster for the endpoint including the Power Source device SHALL include the Grid tag if it is connected to the premises wiring.

The Electrical Sensor device SHALL also conform to the following:

- An Electrical Sensor device SHALL measure the energy and power flows of the Solar Power device at the AC grid or DC connection point.
- If the Solar Power device is connected to AC wiring, this Electrical Power Measurement cluster SHALL support the AlternatingCurrent feature, and SHALL support the PolyPhasePower feature if the Solar Power device is connected via polyphase wiring.
- If the Solar Power device is connected to DC wiring, this Electrical Power Measurement cluster SHALL support the DirectCurrent feature.
- This Electrical Power Measurement cluster SHOULD support the ReactivePower attribute if connected to AC wiring.
- This Electrical Energy Measurement cluster SHOULD support the CumulativeEnergy feature.

If a Solar Power device supports two or three phase power output then it MAY include two or three additional endpoints, each including an Electrical Sensor Device Type as child elements. For each such child endpoint it SHALL include a semantic tag from the Electrical Measurement Namespace in the TagList attribute of the Descriptor cluster to describe the endpoint for the relevant Electrical Power Measurement and Electrical Energy Measurement clusters indicating the relevant AC phase that is being measured.

If a Solar Power device supports measurement of the output of individual solar panels or strings of solar panels then it MAY include additional endpoints for each such measurement, including an Electrical Sensor Device Type as child elements. For each such child endpoint:

- It SHALL include a semantic tag from a Common Namespace, or a Manufacturer defined Tag and Label, in the TagList attribute of the Descriptor cluster to describe the endpoint for the relevant Electrical Power Measurement and Electrical Energy Measurement clusters, indicating the relevant device port, panel, or string of panels that is being measured.
- It SHALL also include a User Label cluster to allow an installer to add identifying information for the panel or string of panels.

If the Solar Power device output power can be controlled, then the Device Energy Management device SHALL be included.

Any Temperature Sensors included SHALL include Tag(s), and for non-standard Namespaces, Label(s) in the Descriptor clusters of their endpoints to identify the temperature being measured.

14.3.6. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

ID	Name	Client/Server	Conformance
0x0003	Identify	Server	O

14.4. Battery Storage Device Type

A Battery Storage device is a device that allows a DC battery, which can optionally be comprised of a set parallel strings of battery packs and associated controller, and an AC inverter, to be monitored and controlled by an Energy Management System in order to manage the peaks and troughs of supply and demand, and/or to optimize cost of the energy consumed in premises. It is not intended to be used for a UPS directly supplying a set of appliances, nor for portable battery storage devices.

14.4.1. Battery Storage Architecture

A Battery Storage device is always defined via endpoint composition. See the Section 14.4.5, “Device Type Requirements” section for more details.

An example of a Battery Storage device with single phase AC output is illustrated below.

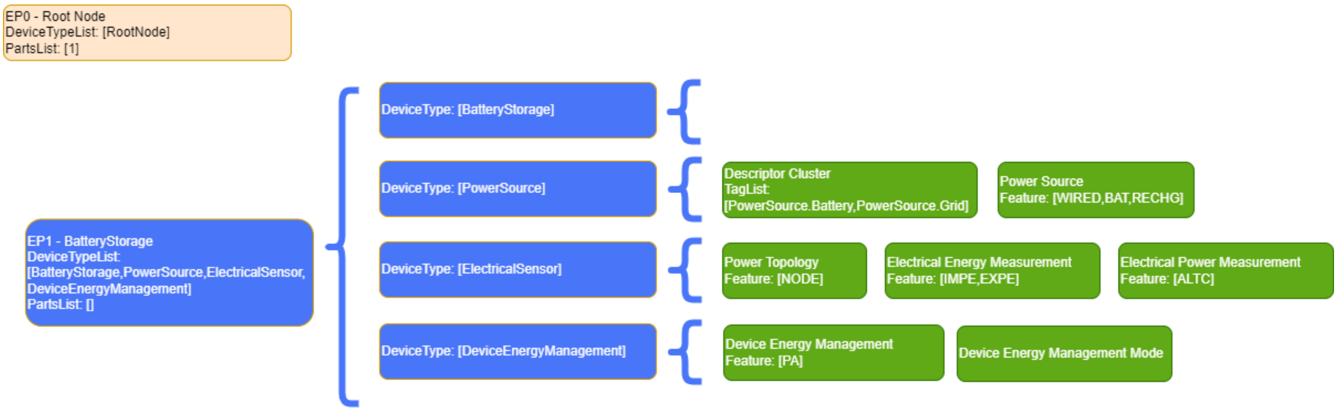

EP0 - Root Node
DeviceTypeList: [RootNode]
PartsList: [1]
EP1 - BatteryStorage
DeviceTypeList:
[BatteryStorage,PowerSource,ElectricalSensor,
DeviceEnergyManagement]
PartsList: []
DeviceType: [BatteryStorage]
DeviceType: [PowerSource]
Descriptor Cluster
TagList:
[PowerSource.Battery,PowerSource.Grid]
Power Source
Feature: [WIRED,BAT,RECHG]
DeviceType: [ElectricalSensor]
Power Topology
Feature: [NODE]
Electrical Energy Measurement
Feature: [IMPE,EXPE]
Electrical Power Measurement
Feature: [ALTC]
DeviceType: [DeviceEnergyManagement]
Device Energy Management
Feature: [PA]
Device Energy Management Mode

Figure 16. Example of a Battery Storage device with single phase AC output

An example of a Battery Storage device which also includes a directly connected Solar Power device supplying DC power to the battery and using a single common inverter to the single phase AC input and output is illustrated below.

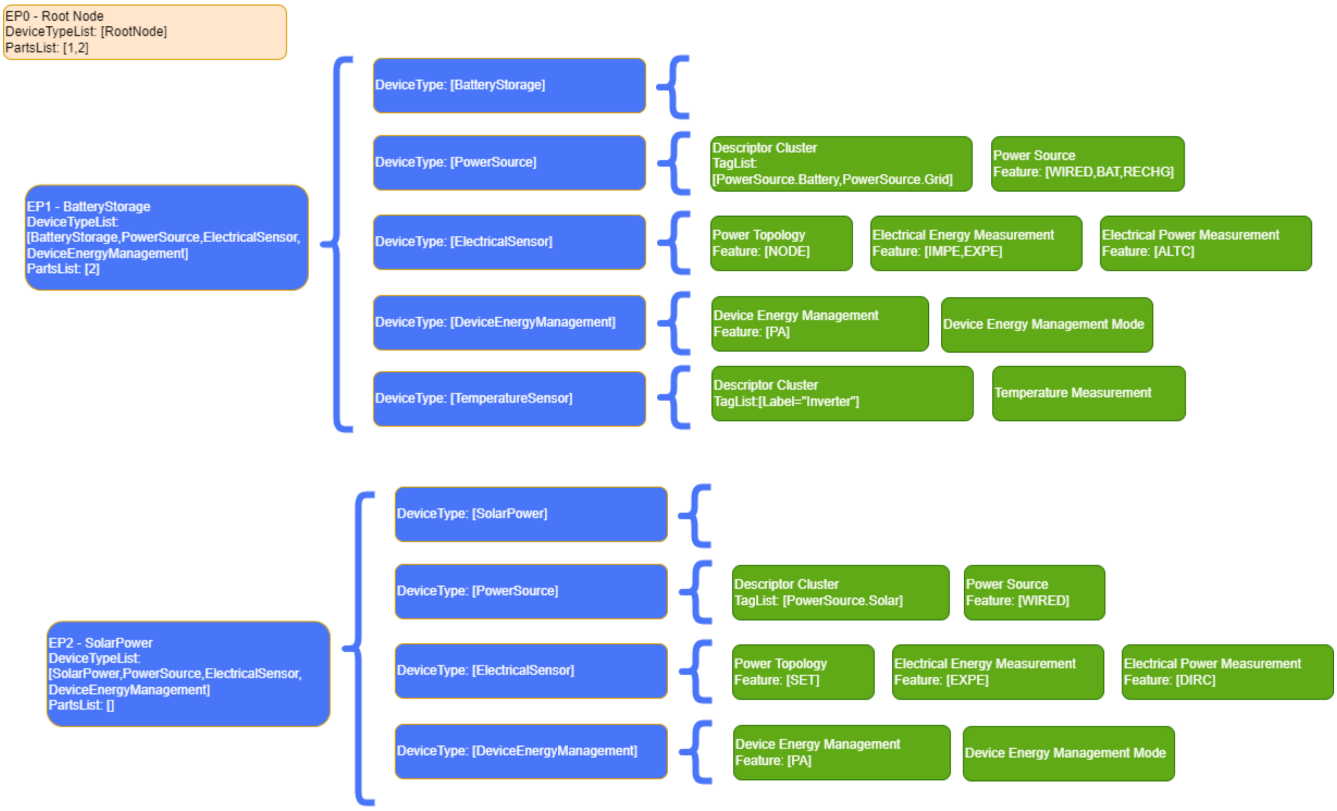

EP0 - Root Node
DeviceTypeList: [RootNode]
PartsList: [1,2]
EP1 - BatteryStorage
DeviceTypeList:
[BatteryStorage,PowerSource,ElectricalSensor,
DeviceEnergyManagement]
PartsList: [2]
DeviceType: [BatteryStorage]
DeviceType: [PowerSource]
Descriptor Cluster
TagList:
[PowerSource.Battery,PowerSource.Grid]
Power Source
Feature: [WIRED,BAT,RECHG]
DeviceType: [ElectricalSensor]
Power Topology
Feature: [NODE]
Electrical Energy Measurement
Feature: [IMPE,EXPE]
Electrical Power Measurement
Feature: [ALTC]
DeviceType: [DeviceEnergyManagement]
Device Energy Management
Feature: [PA]
Device Energy Management Mode
DeviceType: [TemperatureSensor]
Descriptor Cluster
TagList:[Label="Inverter"]
Temperature Measurement
EP2 - SolarPower
DeviceTypeList:
[SolarPower,PowerSource,ElectricalSensor,
DeviceEnergyManagement]
PartsList: []
DeviceType: [SolarPower]
DeviceType: [PowerSource]
Descriptor Cluster
TagList: [PowerSource.Solar]
Power Source
Feature: [WIRED]
DeviceType: [ElectricalSensor]
Power Topology
Feature: [SET]
Electrical Energy Measurement
Feature: [EXPE]
Electrical Power Measurement
Feature: [DIRC]
DeviceType: [DeviceEnergyManagement]
Device Energy Management
Feature: [PA]
Device Energy Management Mode

Figure 17. Example of a Battery Storage device with DC-connected Solar Power device

14.4.2. Revision History

This is the revision history for this document.

Revision	Description
1	Initial revision

14.4.3. Classification

ID	Device Name	Superset Of	Class	Scope
0x0018	Battery Storage		Simple	Endpoint

14.4.4. Conditions

See the Base Device Type definition for conformance tags.

14.4.5. Device Type Requirements

A Battery Storage device SHALL be composed of at least one endpoint with device types as defined by the conformance below. There MAY be more endpoints with additional instances of these device types or additional device types existing in the Battery Storage device.

ID	Name	Constraint	Conformance
0x0011	Power Source	min 1	M
0x0510	Electrical Sensor	min 1	M
0x050D	Device Energy Management		M
0x0302	Temperature Sensor		O
0x0017	Solar Power		O

The Solar Power devices, if included, SHALL have separate endpoints, and include their own Power Source, Electrical Sensor, and Device Energy Management devices, as defined by the Solar Power Device device.

14.4.5.1. Cluster Requirements on Composing Device Types

Below list qualities and conformance that override the cluster specification requirements for the composing device types. A blank table cell means there is no change to that item and the value from the cluster specification applies.

Device ID	Device	Cluster ID	Cluster	Element	Name	Constraint	Access	Conformance
0x0011	Power Source	0x002F	Power Source	Feature	Wired			M
0x0011	Power Source	0x002F	Power Source	Feature	Battery			M
0x0011	Power Source	0x002F	Power Source	Attribute	BatVoltage			M
0x0011	Power Source	0x002F	Power Source	Attribute	BatPercentRemaining			M

Device ID	Device	Cluster ID	Cluster	Element	Name	Constraint	Access	Conformance
0x0011	Power Source	0x002F	Power Source	Attribute	BatTimeRemaining			M
0x0011	Power Source	0x002F	Power Source	Attribute	ActiveBatFaults			M
0x0011	Power Source	0x002F	Power Source	Attribute	BatCapacity			M
0x0011	Power Source	0x002F	Power Source	Attribute	BatTimeToFullCharge			M
0x0011	Power Source	0x002F	Power Source	Attribute	BatChargingCurrent			M
0x0011	Power Source	0x002F	Power Source	Attribute	ActiveBatChargeFaults			M
0x0011	Power Source	0x001D	Descriptor	Feature	TagList			M
0x0510	Electrical Sensor	0x0090	Electrical Power Measurement					M
0x0510	Electrical Sensor	0x0090	Electrical Power Measurement	Feature	AlternatingCurrent			M
0x0510	Electrical Sensor	0x0090	Electrical Power Measurement	Attribute	Voltage			M
0x0510	Electrical Sensor	0x0090	Electrical Power Measurement	Attribute	ActiveCurrent			M
0x0510	Electrical Sensor	0x0091	Electrical Energy Measurement					M

Device ID	Device	Cluster ID	Cluster	Element	Name	Constraint	Access	Conformance
0x0510	Electrical Sensor	0x0091	Electrical Energy Measurement	Feature	ExportedEnergy			M
0x050D	Device Energy Management	0x0098	Device Energy Management	Feature	PowerAdjustment			M
0x0302	Temperature Sensor	0x001D	Descriptor	Feature	TagList			M

The Power Source cluster in the Power Source device SHALL support the RECHG feature if it can be charged as well as discharged through the connection to the premises wiring.

The Electrical Sensor device SHALL also conform to the following:

- An Electrical Sensor device SHALL measure the energy and power flows of the Battery Storage device at the AC grid connection point.
- The Electrical Power Measurement cluster of this Electrical Sensor device SHALL support the PolyphasePower feature if the Battery Storage device is connected via polyphase wiring, and SHOULD support the ReactivePower attribute.
- The Electrical Energy Measurement cluster of this Electrical Sensor device SHALL support the ImportedEnergy feature if it can be charged as well as discharged through the connection to the premises wiring, and SHOULD support the CumulativeEnergy feature.

If a Battery Storage device supports two or three phase power output then it MAY include two or three additional endpoints, each including an Electrical Sensor Device Type as child elements. For each such child endpoint it SHALL include a semantic tag from the Electrical Measurement Namespace in the TagList attribute of the Descriptor cluster to describe the endpoint for the relevant Electrical Power Measurement and Electrical Energy Measurement clusters indicating the relevant AC phase that is being measured.

If a Battery Storage device supports measurement of the input and output of individual batteries or sets of batteries then it MAY include additional endpoints for each such measurement, including an Electrical Sensor Device Type as child elements. For each such child endpoint: * it SHALL include a semantic tag from the Common Number Namespace, or a Manufacturer defined Tag and Label, in the TagList attribute of the Descriptor cluster to describe the endpoint for the relevant Electrical Power Measurement and Electrical Energy Measurement clusters indicating the relevant device port, battery, or set of batteries that is being measured. * it SHOULD also include a User Label cluster to allow an installer to add identifying information if the device permits flexible connection of the actual batteries at installation time.

Any Temperature Sensors included SHALL include Tag(s), and for non-standard Namespaces, Label(s) in the Descriptor clusters of their endpoints to identify the temperature being measured.

14.4.6. Semantic Tag Requirements on Composing Device Types

The table below lists conformance requirements for semantic tags associated with the device types listed in the table when used in a composition.

Device ID	Device	Namespace ID	Namespace	Tag ID	Tag	Conformance
0x0011	Power Source	0x0F	Power Source	0x01	Grid	M
0x0011	Power Source	0x0F	Power Source	0x03	Battery	M

14.4.7. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

ID	Name	Client/Server	Conformance
0x0003	Identify	Server	O

14.5. Heat Pump Device Type

A Heat Pump device is a device that uses electrical energy to heat either spaces or water tanks using ground, water or air as the heat source. These typically can heat the air or can pump water via central heating radiators or underfloor heating systems. It is typical to also heat hot water and store the heat in a hot water tank.

Note that the Water Heater device type can also be heated by a heat pump and has similar requirements, but that cannot be used for space heating.

14.5.1. Heat Pump Architecture

A Heat Pump device is always defined via endpoint composition. See the Section 14.5.5, “Device Type Requirements” section for more details.

The Heat Pump device may contain Temperature Sensors for example to measure the flow and return temperatures of the water it is providing to the premises heating system.

The Heat Pump device may also include Thermostats located in the rooms that are being heated by it, which in turn may also include Temperature Sensor clusters which can report the temperatures in those rooms. These Thermostats may be included as servers within Thermostat devices within the Heat Pump device itself, or may be separate third-party Thermostat devices for which the Heat Pump has a client to use them.

An example of a Heat Pump device is illustrated below.

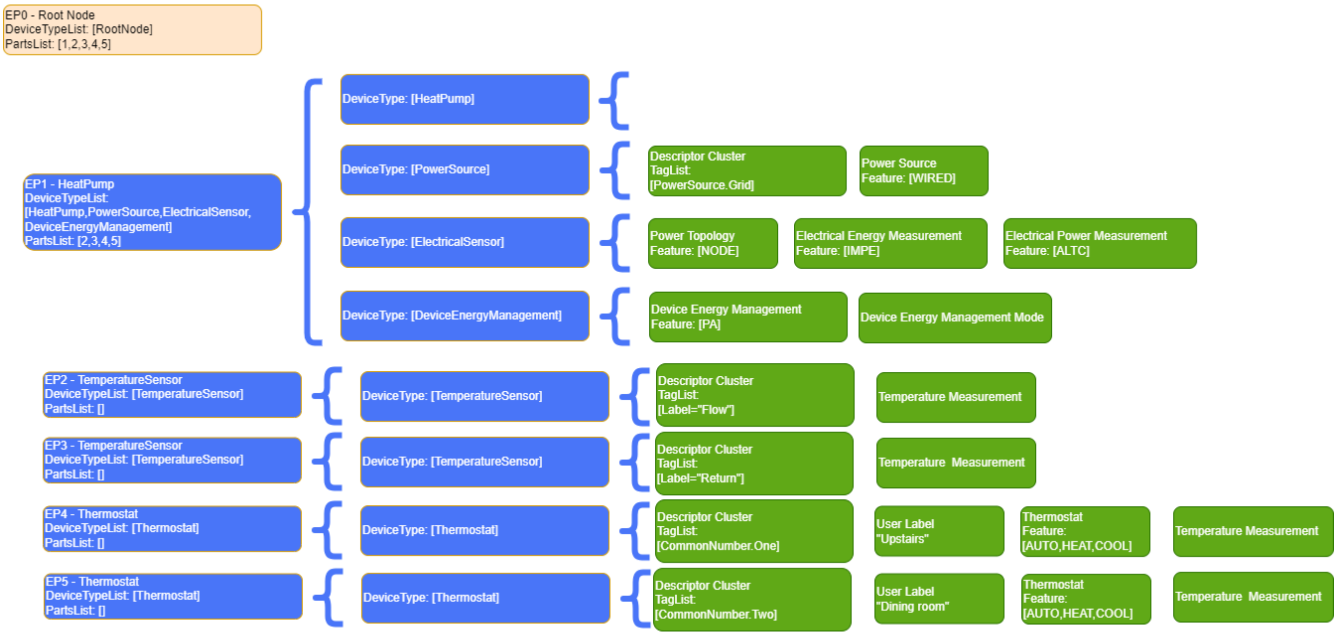


Figure 18. Example of a Heat Pump device

14.5.2. Revision History

This is the revision history for this document.

Revision	Description
1	Initial revision

14.5.3. Classification

ID	Device Name	Superset Of	Class	Scope
0x0309	Heat Pump		Simple	Endpoint

14.5.4. Conditions

See the Base Device Type definition for conformance tags.

14.5.5. Device Type Requirements

A Heat Pump device SHALL be composed of at least one endpoint with device types as defined by the conformance below. There MAY be more endpoints with additional instances of these device types or additional device types existing in the Heat Pump device.

ID	Name	Constraint	Conformance
0x0011	Power Source		M
0x0510	Electrical Sensor	min 1	M
0x050D	Device Energy Management		M
0x0301	Thermostat		O

ID	Name	Constraint	Conformance
0x050F	Water Heater		O
0x0302	Temperature Sensor		O

The Heat Pump device SHALL include either one or more Thermostat devices, or include a Thermostat client.

14.5.5.1. Cluster Requirements on Composing Device Types

Below list qualities and conformance that override the cluster specification requirements for the composing device types. A blank table cell means there is no change to that item and the value from the cluster specification applies.

Device ID	Device	Cluster ID	Cluster	Element	Name	Constraint	Access	Conformance
0x0011	Power Source	0x002F	Power Source	Feature	Wired			M
0x0011	Power Source	0x001D	Descriptor	Feature	TagList			M
0x0510	Electrical Sensor	0x0090	Electrical Power Measurement					M
0x0510	Electrical Sensor	0x0090	Electrical Power Measurement	Feature	AlternatingCurrent			M
0x0510	Electrical Sensor	0x0090	Electrical Power Measurement	Attribute	Voltage			M
0x0510	Electrical Sensor	0x0090	Electrical Power Measurement	Attribute	ActiveCurrent			M
0x0510	Electrical Sensor	0x0091	Electrical Energy Measurement					M
0x050D	Device Energy Management	0x0098	Device Energy Management	Feature	PowerAdjustment			M

Device ID	Device	Cluster ID	Cluster	Element	Name	Constraint	Access	Conformance
0x0301	Thermostat	0x001D	Descriptor	Feature	TagList			M
0x0301	Thermostat	0x0041	User Label					M
0x0302	Temperature Sensor	0x001D	Descriptor	Feature	TagList			M

If a Heat Pump device supports two or three phase power input then it MAY include two or three additional endpoints, each including an Electrical Sensor Device Type as child elements. For each such child endpoint it SHALL include a semantic tag from the Electrical Measurement Namespace in the TagList attribute of the Descriptor cluster to describe the endpoint for the relevant Electrical Power Measurement and Electrical Energy Measurement clusters indicating the relevant AC phase that is being measured.

The Electrical Energy Measurement and Electrical Power Measurement clusters of the mandatory Electrical Sensor device SHALL measure the energy and power of the Heat Pump device at the AC grid connection point.

The Electrical Power Measurement clusters of the mandatory Electrical Sensor device SHALL support the PolyPhasePower feature if the Heat Pump device is connected via polyphase wiring.

The Electrical Energy Measurement cluster of the mandatory Electrical Sensor device SHOULD support the ImportedEnergy and CumulativeEnergy features.

Any Temperature Sensors using non-standard Namespaces for their Tags SHALL include Label(s) in the Descriptor clusters of their endpoints to identify the temperature being measured.

Any Thermostat SHALL include a semantic tag from a Common Namespace, or a Manufacturer defined Tag and Label, in the TagList attribute of the Descriptor cluster to describe the endpoint for the relevant Thermostat clusters, indicating the relevant device (or its connected port) that is being measured. It SHALL also include a User Label cluster to allow an installer to add identifying information for the rooms or spaces where the Thermostat measurement point is located.

14.5.6. Semantic Tag Requirements on Composing Device Types

The table below lists conformance requirements for semantic tags associated with the device types listed in the table when used in a composition.

Device ID	Device	Namespace ID	Namespace	Tag ID	Tag	Conformance
0x0011	Power Source	0x0F	Power Source	0x01	Grid	M

14.5.7. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

ID	Name	Client/Server	Conformance
0x0003	Identify	Server	O
0x0201	Thermostat	Client	O

Chapter 15. Network Infrastructure Device Types

15.1. Introduction

Matter aims to build a universal IPv6-based communication protocol for smart home devices, and in principle almost any IPv6-bearing Wi-Fi, Thread, or Ethernet network is suitable for Matter deployment (see *Network Topology* in [MatterCore]).

The Network Infrastructure device types defined in this chapter are designed to provide a reliable, secure, and performant connectivity experience for Matter devices and their users. To achieve this, these device types expose Matter application clusters for the purposes of network management, diagnostics, and configuration. They also include requirements on lower layers (including the network and physical layers) that contribute to the desired connectivity experience.

Note that Matter devices and applications cannot rely on the presence of Matter-certified Network Infrastructure Devices, or the presence of network features mandated by these device types, for their correct operation.

15.2. Common Requirements

15.2.1. Minimum Number of Devices to Support

The count of IoT devices within a smart home can quickly grow to a substantial number when factoring in the quantity of light points, light switches, and an array of sensor devices that all require connectivity. Consequently, the network infrastructure must be capable of accommodating a substantial number of devices to avoid hitting constraints when users wish to add "one more" product. The requirements in this chapter were designed to result in an IP network (which MAY be a combination of Ethernet LAN, Wi-Fi WLAN, and Thread PAN networks) with a minimum of 300 Matter devices.

- Requirements related to Wi-Fi Access Points and Thread Border Routers are detailed in the sections for the Network Infrastructure Manager and Thread Border Router device types below.

15.3. Network Infrastructure Manager Device Type

A Network Infrastructure Manager provides interfaces that allow for the management of the Wi-Fi, Thread, and Ethernet networks underlying a Matter deployment, realizing the *Star Network Topology* described in [MatterCore].

Examples of physical devices that implement the Matter Network Infrastructure Manager device type include Wi-Fi gateway routers.

Relevant hardware and software requirements for Network Infrastructure Manager devices are defined in Section 15.3.6, “Other Requirements” and within the clusters mandated by this device type.

A Network Infrastructure Manager device MAY be managed by a service associated with the device vendor, for example, an Internet Service Provider. Sometimes this managing service will have policies that require the use of the `Managed Device` feature of the `Access Control Cluster` (see Section 15.3.5.1, “Access Control MNGD Conformance”). Consequently, Commissioners of this device type should be aware of this feature and its use.

15.3.1. Revision History

This is the revision history for this document.

Revision	Description
1	Initial release
2	Add Thread Network Diagnostics requirement; extend "Other Requirements" section

15.3.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x0090	Network Infrastructure Manager		Simple	Endpoint

15.3.3. Conditions

See the Base Device Type definition for conformance tags.

15.3.4. Cluster Requirements

Each endpoint supporting this device type SHALL include these clusters based on the conformance defined below.

ID	Name	Client/Server	Quality	Conformance
0x0451	Wi-Fi Network Management	Server		M
0x0452	Thread Border Router Management	Server		M
0x0453	Thread Network Directory	Server		M
0x0035	Thread Network Diagnostics	Server		M

15.3.5. Root Node Element Requirements

Below list qualities and conformance that override the cluster specification requirements for the Section 2.1, “Root Node”. A blank table cell means there is no change to that item and the value from the cluster specification applies.

ID	Cluster	Element	Name	Constraint	Access	Conformance
0x001F	Access Control	Feature	MNGD			desc

15.3.5.1. Access Control MNGD Conformance

A Network Infrastructure Manager device MAY utilize the Managed Device (MNGD) feature flag of the Access Control Cluster on the device’s Root Node endpoint (i.e. Endpoint 0).

Please refer to the "Managed Device Feature Usage Restrictions" section in the Access Control Cluster chapter of the Matter Core Specification for the complete set of limitations on use of this feature on endpoints with the Network Infrastructure Manager device type.

NOTE The conformance of this element crosses endpoints. It is expressed against the Root Node endpoint and there SHALL NOT be a separate AccessControl cluster on the endpoint having the Network Infrastructure Manager device type.

15.3.6. Other Requirements

The Network Infrastructure Manager SHALL implement a bridged Wi-Fi / Ethernet hub network, enabling IPv6 connectivity between Matter Nodes across transports.

The Network Infrastructure Manager SHALL support IP communication with at least 300 Matter devices on this network. If the Network Infrastructure Manager operates a DHCPv4 server, then it SHOULD be configured by default with a subnet mask and DHCP pool size that allow for at least 300 devices.

NOTE This recommendation is meant to avoid IPv4 address exhaustion if a large number of devices request IPv4 addresses e.g. for non-Matter traffic.

15.3.6.1. Ethernet Requirements

The device SHALL provide an Ethernet LAN interface that is part of the bridged hub network.

The Root Node endpoint of the device MAY include a Network Commissioning cluster associated with this Ethernet interface.

15.3.6.2. Wi-Fi Requirements

The device SHALL support the operation of an IEEE 802.11 Wi-Fi network (ESS) and provide access to the SSID and credentials of this network via the Wi-Fi Network Management cluster. The mechanisms by which this network is configured are outside the scope of this specification. The device SHALL support concurrently operating BSSs for this ESS in the 2.4 GHz and 5 GHz frequency bands; it MAY support operating additional BSSs for this ESS in other frequency bands such as 6 GHz or sub-1 GHz. All BSSs in this ESS SHALL be part of the bridged hub network, i.e. bridged to each other and to the Ethernet interface.

The device MAY support operating additional Wi-Fi networks (e.g. a "guest network"); the requirements above do not apply to any such additional networks.

The device SHOULD NOT include any Network Commissioning cluster instances associated with the Wi-Fi Access Point interface.

The device SHALL be certified by the Wi-Fi Alliance in the Access Point role for Wi-Fi 6 or above. It SHALL additionally be certified in the Access Point role for Wi-Fi 6E if it supports operating in the 6 GHz band, and for Wi-Fi HaLow if it supports operating in the sub-1 GHz band.

To support the efficient operation of the network generally, and for low-power stations in particular, the Network Infrastructure Manager SHALL support, and upon (or before) Matter commissioning SHALL enable, the following Wi-Fi features:

- Extended Sleep Time with a sleep time support up to at least 60 minutes, which includes the following IEEE 802.11 features:
    - Basic Service Set (BSS) Max Idle Period
    - dot11BSSMaxIdlePeriodIndicationByNonAPSTA
- IPv6 Proxy Neighbor Discovery Protocol (NDP) including IPv6 duplicate address detection
- IPv4 Proxy Address Resolution Protocol (ARP)

The device SHALL support at least 100 simultaneous Wi-Fi associations - irrespective of the distribution of the Matter Wi-Fi devices over the supported bands; all Matter Wi-Fi devices could be on the same band. This includes associations of low-power Matter Wi-Fi devices that are asleep for a long time.

NOTE For the case of in-field upgrades of pre-Matter Wi-Fi access points, an exemption (simultaneous association requirement reduced to 64) can be requested when applying for Matter certification, e.g. in case of Wi-Fi chipset limitations (see the Alliance Certification Policy).

15.3.6.3. Thread Requirements

The Network Infrastructure Manager device SHALL be certified by the Thread Group as *Built on Thread: Border Router* based on Thread 1.4.0 or above.

The device SHALL implement a Thread Border Router as described by the Thread specification, and provide connectivity between the Thread network and the Wi-Fi / Ethernet hub network. The Thread Interface associated with the Border Router SHALL be exposed via the Thread Border Router Management cluster.

The Thread Network Diagnostics cluster included on the endpoint SHALL be the instance corresponding to the Thread Interface associated with the Border Router functionality.

The device SHOULD NOT include any Network Commissioning cluster instances associated with the Thread Border Router.

The Network Infrastructure Manager device SHALL support working as a Thread Parent for a minimum of 64 Thread Children simultaneously in any combination of Children End Device types. The Network Infrastructure Manager device SHALL support operating as a Thread Border Router in any Thread Network with up to 150 Thread nodes.

15.3.6.4. Discovery and Commissioning

A Network Infrastructure Manager device SHOULD implement Extended Discovery in order to be discoverable by entities on the local IP network, even when not in Commissioning Mode, and SHOULD populate the optional *device type* subtype (e.g., _T144) to allow for filtering of discovery results to find only Nodes that match the Network Infrastructure Manager device type (see Commissioning Subtypes).

A Network Infrastructure Manager SHOULD populate the following DNS-SD TXT record key/value pairs in the Commissionable Node Discovery response: Commissioning Pairing Hint, and Commissioning Pairing Instruction so that the Commissioner can guide the user through the steps needed to put the Commissionee into Commissioning Mode. If the Network Infrastructure Manager provides its own app or website which includes a UX for putting the device into Commissioning Mode, then the device SHOULD populate the Commissioning VID/PID key/value pair and SHOULD set bit 1 of the Pairing Hint (Device Manufacturer URL), so that the Commissioner can utilize the URL specified in the CommissioningCustomFlowUrl of the DeviceModel schema entry indexed by the Vendor ID and Product ID in the Distributed Compliance Ledger and utilize flows described in Custom Commissioning Flow to redirect the user to a custom app or website specified by the device vendor, and receive the user back following the callback flow which contains the onboarding payload. This flow is described in detail in the Initiating Commissioning section of the Matter Core specification, under the User Journey titled User-Initiated Beacon Detection, Already Commissioned Device.

15.4. Thread Border Router Device Type

A Thread Border Router device type provides interfaces for querying and configuring the associated Thread network.

Instances of physical devices categorized as Thread Border Routers encompass standalone Thread Border Routers, conventional application devices like smart speakers, media streamers, and lighting fixtures equipped with a Thread Border Router, as well as Wi-Fi Routers incorporating Thread Border Router functionality.

The necessary hardware and software prerequisites are detailed within the clusters that are mandated by this device type.

15.4.1. Revision History

This is the revision history for this document.

Revision	Description
1	Initial revision
2	Extend "Other Requirements" section

15.4.2. Classification

ID	Device Name	Superset Of	Class	Scope
0x0091	Thread Border Router		Simple	Endpoint

15.4.3. Conditions

See the Base Device Type definition for additional conformance tags.

15.4.4. Device Type Requirements

The following table lists other device types to be implemented along with this device type based on conformance.

ID	Name	Constraint	Conformance
0x0019	Secondary Network Interface		O

If a Thread Border Router endpoint supports the Secondary Network Interface device type, then

- The Thread Border Router Management cluster and the Network Commissioning Cluster SHALL reflect the same underlying network configuration, i.e. changes made via either cluster SHALL also be reflected in the other.
- The MaxNetworks attribute in the Network Commissioning Cluster SHALL have a value of 1.

15.4.5. Cluster Requirements

Each endpoint supporting the Thread Border Router device type SHALL include these clusters based on the conformance defined below.

ID	Name	Client/Server	Quality	Conformance
0x0035	Thread Network Diagnostics	Server		M
0x0452	Thread Border Router Management	Server		M
0x0453	Thread Network Directory	Server		O

15.4.6. Other Requirements

The device SHALL implement a Thread Border Router as described by the Thread specification, and provide connectivity between the Thread network and a hub network when connected via a functioning adjacent infrastructure link.

The Thread Interface associated with the Border Router SHALL be exposed via the Thread Border Router Management cluster.

The device MAY include a Network Commissioning cluster instance associated with the Wi-Fi or Ethernet adjacent infrastructure link interface on its Root Node. It SHOULD NOT include a Network Commissioning cluster instance associated with the Thread interface of the Border Router on its Root Node.

15.4.6.1. Thread Requirements

A device exposing the Thread Border Router device type SHALL be certified by the Thread Group as *Built on Thread: Border Router* based on Thread 1.4.0 or above.

The device SHALL implement a Thread Border Router as described by the Thread specification, and provide connectivity between the Thread network and the Wi-Fi / Ethernet hub network. The Thread Interface associated with the Border Router SHALL be exposed via the Thread Border Router Management cluster.

The Thread Border Router device SHALL support working as a Thread Parent for a minimum of 64 Thread Children simultaneously in any combination of Children End Device types. The Thread Border Router device SHALL support operating as a Thread Border Router in any Thread Network with up to 150 Thread nodes.

15.4.7. Cluster Usage

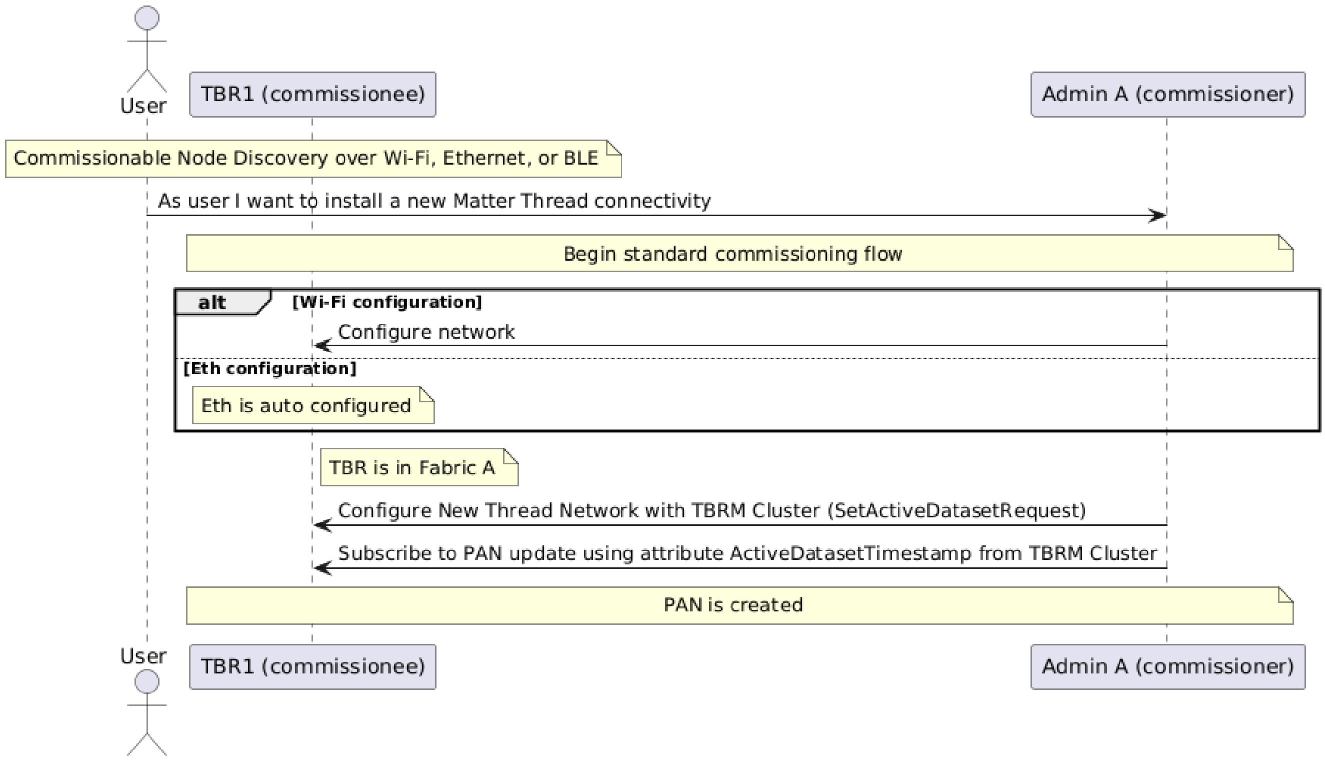
This section describes how to control and monitor the operation of a Thread Border Router.

The Thread Border Router Device Type provides the Thread Border Router Management Cluster with the goal of ensuring a seamless user experience when adding a new Border Router to form a new Thread network or join an existing one. This section presents informative configuration sequences that a Fabric Admin can set up to improve coverage and Internet access redundancy for Thread networks using the Thread Border Router device type. Four use cases are covered:

- Initial configuration of a Thread Border Router when no PAN exists
- Joining an existing PAN
- Sharing an existing PAN
- Moving to a new PAN

15.4.7.1. Initial configuration of Thread Border Router

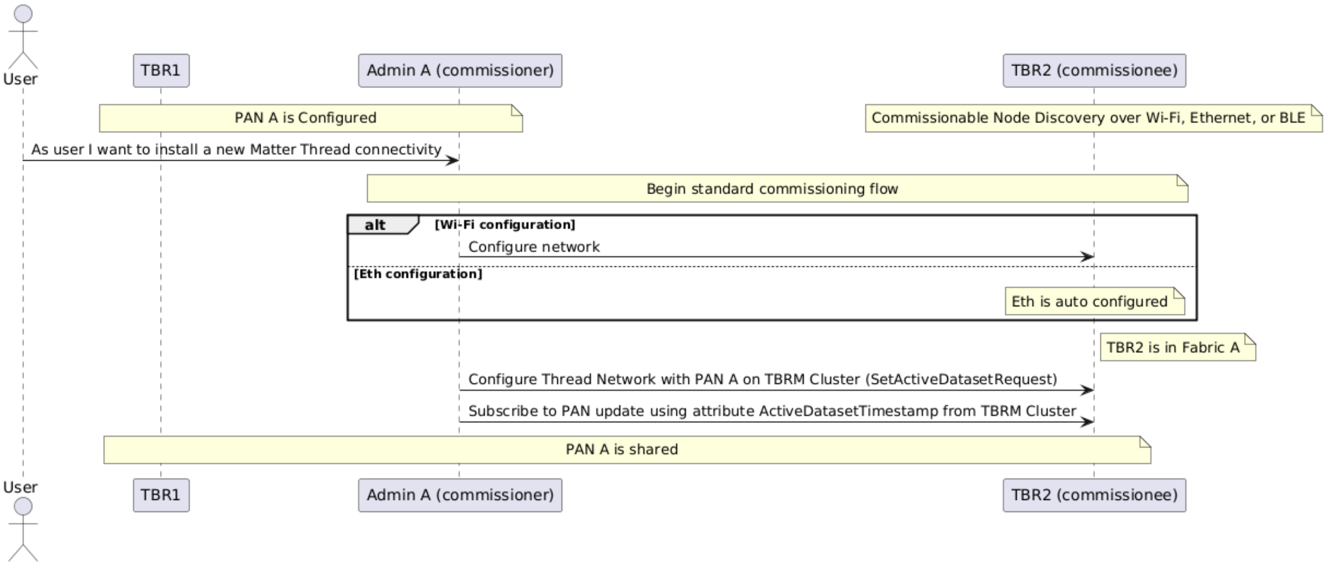
In this use case, there is initially no PAN at the user's home. The user installs a Matter certified Thread Border Router to use Thread connectivity for their Matter devices.


User
TBR1 (commissionee)
Admin A (commissioner)
Commissionable Node Discovery over Wi-Fi, Ethernet, or BLE
As user I want to install a new Matter Thread connectivity
Begin standard commissioning flow
alt
[Wi-Fi configuration]
Configure network
[Eth configuration]
Eth is auto configured
TBR is in Fabric A
Configure New Thread Network with TBRM Cluster (SetActiveDatasetRequest)
Subscribe to PAN update using attribute ActiveDatasetTimestamp from TBRM Cluster
PAN is created
User
TBR1 (commissionee)
Admin A (commissioner)

After installation, the PAN configured by Admin A on the Thread Border Router is used to install the Thread Matter device on Fabric A.

15.4.7.2. Joining an existing PAN

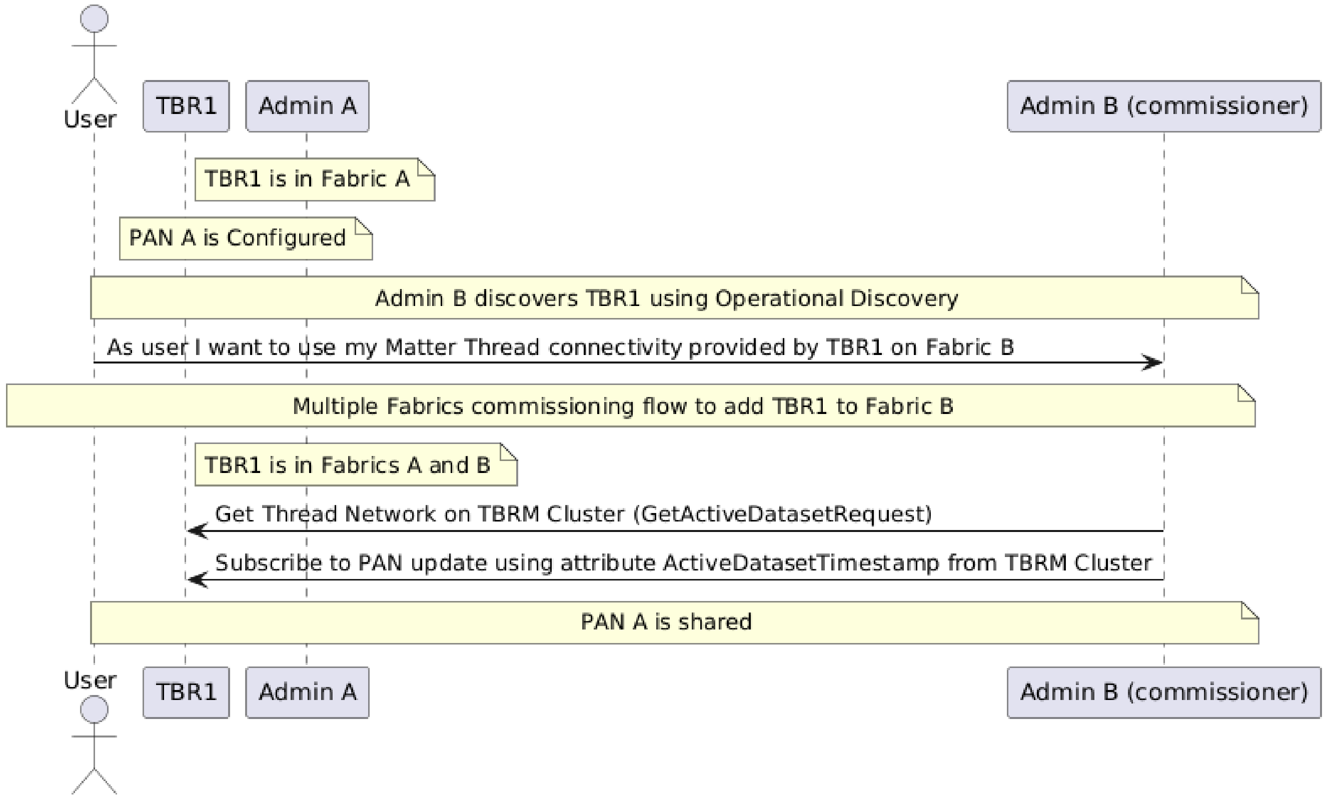
In this use case, there is already a PAN in the user’s home that is managed by a Fabric A with TBR1 (e.g. as a result of the above sequence diagram applied previously between Admin 1 and TBR1). The user installs a Matter-certified Thread Border Router (TBR2) to extend the Thread coverage and Internet access redundancy of Thread networks for their Matter devices.


User
TBR1
Admin A (commissioner)
TBR2 (commissionee)
PAN A is Configured
Commissionable Node Discovery over Wi-Fi, Ethernet, or BLE
As user I want to install a new Matter Thread connectivity
Begin standard commissioning flow
alt
[Wi-Fi configuration]
Configure network
[Eth configuration]
Eth is auto configured
TBR2 is in Fabric A
Configure Thread Network with PAN A on TBRM Cluster (SetActiveDatasetRequest)
Subscribe to PAN update using attribute ActiveDatasetTimestamp from TBRM Cluster
PAN A is shared
User
TBR1
Admin A (commissioner)
TBR2 (commissionee)

After installation, a single Thread PAN is shared by all Border Routers.

15.4.7.3. Share an existing PAN

In this use case, there is already a PAN in the user’s home that is managed by Fabric A with TBR1 (e.g. as a result of the above sequence diagram applied previously between Admin 1 and TBR1). The user wants to use the Thread connectivity provided by the Matter-certified Thread Border Router managed by Fabric A with Fabric B to share its connectivity.


User
TBR1
Admin A
Admin B (commissioner)
TBR1 is in Fabric A
PAN A is Configured
Admin B discovers TBR1 using Operational Discovery
As user I want to use my Matter Thread connectivity provided by TBR1 on Fabric B
Multiple Fabrics commissioning flow to add TBR1 to Fabric B
TBR1 is in Fabrics A and B
Get Thread Network on TBRM Cluster (GetActiveDatasetRequest)
Subscribe to PAN update using attribute ActiveDatasetTimestamp from TBRM Cluster
PAN A is shared
User
TBR1
Admin A
Admin B (commissioner)

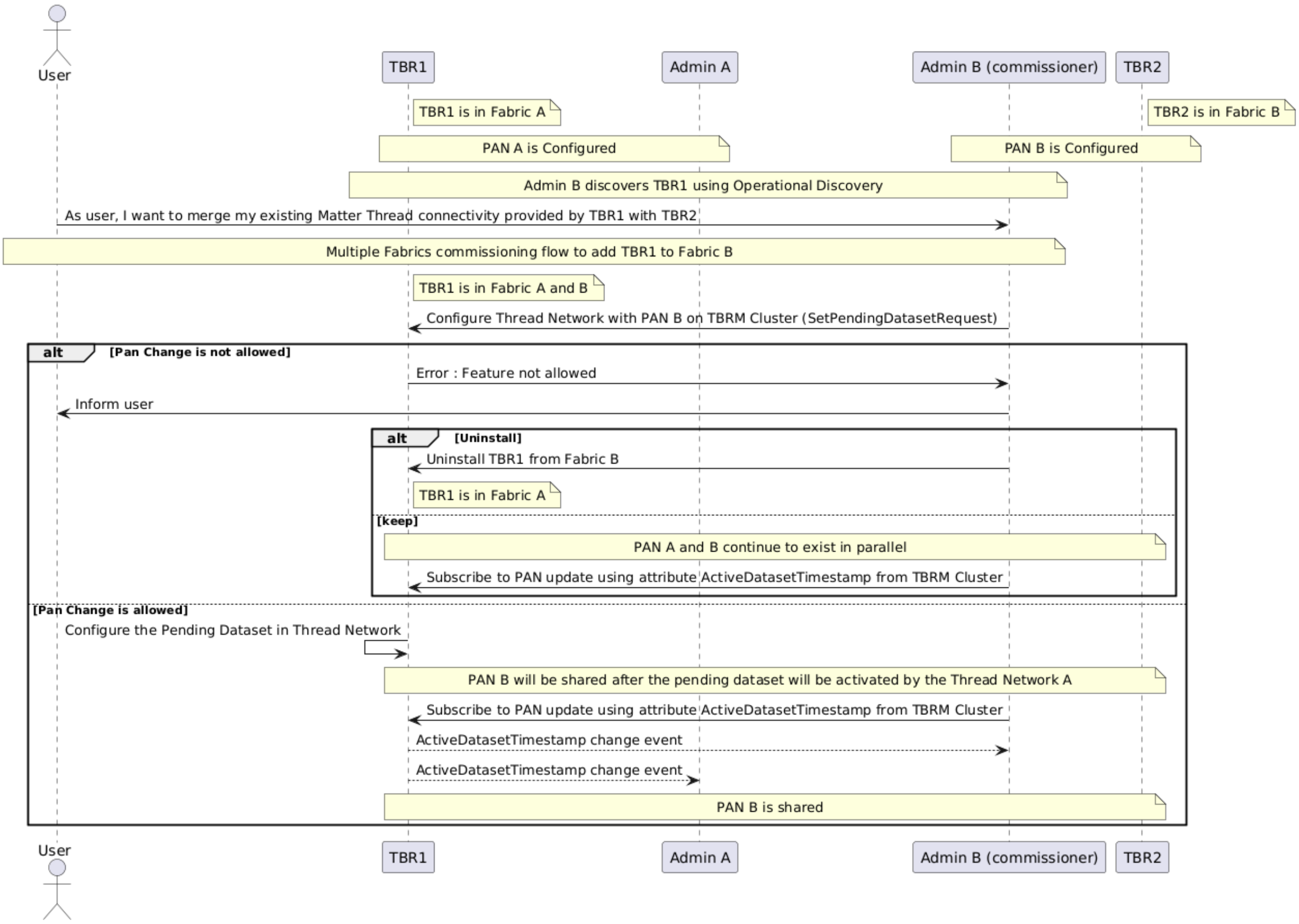
After the installation, the Thread PAN is shared by Fabric A and B.

NOTE The user must use the Multiple Fabrics feature process between Fabrics A and B to commission the Thread Border Router from Fabric A to Fabric B.

15.4.7.4. Merging to a new PAN

This use case is a specific configuration where the user already has 2 PANs. One is managed by a Fabric A through a Matter Thread Border Router (TBR1) and the other is managed by a Fabric B through another Matter Thread Border Router (TBR2).

The user wishes to merge the existing Thread connectivity provided by the two Matter certified Thread Border Routers to share their connectivity to extend the Thread coverage and Internet access redundancy of the Thread networks for their Matter devices.


User
TBR1
Admin A
Admin B (commissioner)
TBR2
TBR1 is in Fabric A
TBR2 is in Fabric B
PAN A is Configured
PAN B is Configured
Admin B discovers TBR1 using Operational Discovery
As user, I want to merge my existing Matter Thread connectivity provided by TBR1 with TBR2
Multiple Fabrics commissioning flow to add TBR1 to Fabric B
TBR1 is in Fabric A and B
Configure Thread Network with PAN B on TBRM Cluster (SetPendingDatasetRequest)
alt
[Pan Change is not allowed]
Error : Feature not allowed
Inform user
alt
[Uninstall]
Uninstall TBR1 from Fabric B
TBR1 is in Fabric A
[keep]
PAN A and B continue to exist in parallel
Subscribe to PAN update using attribute ActiveDatasetTimestamp from TBRM Cluster
[Pan Change is allowed]
Configure the Pending Dataset in Thread Network
PAN B will be shared after the pending dataset will be activated by the Thread Network A
Subscribe to PAN update using attribute ActiveDatasetTimestamp from TBRM Cluster
ActiveDatasetTimestamp change event
ActiveDatasetTimestamp change event
PAN B is shared
User
TBR1
Admin A
Admin B (commissioner)
TBR2

After installation, the Thread PAN is shared between Fabrics A and B.

Note 1 : Activation of the pending dataset is Thread stack dependent and propagation of the new dataset to all Matter devices may fail, for example if devices are disconnected during the process.

Note 2 : This configuration should remain very rare if the previous installations followed the previous use cases.