UM2977

User manual

STM32WB Series Zigbee Cluster Library API

Introduction

Clusters are related sets of commands and attributes dedicated to a specific type of device, service, or sector. Developers use clusters to build a collection of functions suited to their needs. For example, the On/Off cluster allows applications to turn devices on and off, and contains commands and attributes to support that functionality. A developer uses the On/Off cluster in conjunction with an application to allow a device to be activated or deactivated over a Zigbee® network.

This document details the API for Zigbee® 3.0 cluster primitives, or "stubs", which Exegin has implemented based on the Zigbee® Cluster Library 8 specification [1] and [3]. These cluster stubs must be paired with and modified for an application before they are considered functional clusters.

For more information about Zigbee® PRO network protocol, see [1].

For more information about cluster architecture and operations, see [2].

1 General information

This document applies to the STM32WB Series dual-core Arm®-based microprocesor.

Note: Arm is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

Warning: *This document, rev 1.0, has been issued before the release of the STM32WB V1.13 Cube firmware package.*

There may be some minor misalignment between this document and the actual firmware package content.

1.1 References

Table 1. **Reference documents**

Name	Title
[1]	05-3474-22, Zigbee Specification R22
[2]	07-5123-08, Zigbee Cluster Library Specification 8
[3]	The Zigbee Smart Energy Standard

2 Overview

This section provides basic information about cluster architecture and operation, covering such topics as:

- Servers and clients
- Attributes and commands
- Cluster pointers
- Allocation functions
- Endpoints

2.1 Client / server relationship

The cluster operation follows a client / server model with clusters having a client component and a server component.

The server manages the communication and interaction with the Zigbee® device, transmitting any client instructions in line with the cluster attribute values. It also reports to the client on changes to the attribute values resulting from any external events.

In the case of the On/Off cluster, the server side of the On/Off cluster is hosted by a light fixture for example and receives "on" or "off" commands from the client side of the cluster hosted by a light switch.

The client is responsible for altering the state of the cluster attributes on the server by sending commands to the server, or for reporting the state of the server attributes to its application. In the case of the On/Off cluster example, when the light switch is toggled, it sends an "on" or "off" command to the light or lights which it is bound to. In turn, the lights might send notifications of the change of their state attributes back to the switch.

Clients and servers are linked through bindings when they are created. Clusters have unique identifiers used for communicating.

For more information about the client / server relationship, see Section 2.2.2 in [1].

2.2 Attributes and commands

In general:

- Attributes are variables representing the current state of a device. They are commonly held by the server portion of a cluster.
- Commands are functions used by applications to alter or report the attribute values. They are commonly generated by the client portion and sent to the server.

As an example:

- The On/Off cluster defines the functionality for devices that can be turned on or off. The On/Off cluster contains the On/Off attribute, the state which determines if a device is "on" or "off"
- The On/Off cluster defines commands that read or change the state of the On/Off attribute

Commands are divided into two types:

- Cluster specific (unique to a cluster)
- Profile wide (available on every cluster)

While profile wide commands are intended to access a cluster attributes, each cluster has its own unique cluster specific commands. Attributes are primarily accessed through profile-wide commands such as:

- Read attribute
- Write attribute

Profile-wide commands also include other general functions such as default response, configure reporting, discover attributes, and so on. Profile-wide commands are the same for all clusters.

For more information about attributes and commands, see Section 2 of [1].

2.3 Endpoints

Endpoints are a central feature of the cluster addressing and usually represent a discrete logical device such as a light switch. Only a single instance of a cluster is allowed on a given endpoint, but each endpoint typically supports multiple clusters, (such as Basic cluster, Alarms cluster, Level cluster, and so on).

An application creates endpoints using the `ZbZclAddEndpoint()` function, which is declared in the Zigbee® cluster library header file, `zcl.h`. When creating a new endpoint, the application must select a new unique endpoint between 1 and 239, and provide the `ProfileId` and `DeviceID`. Endpoints outside of the 1-239 range are reserved for internal Zigbee® functions, such as the Zigbee® Device object on endpoint 0.

After creating an endpoint, the application creates application clusters and adds them to that endpoint. By default, every endpoint includes the Basic cluster.

When an endpoint is created, the stack internally creates a simple descriptor for each endpoint created. The simple descriptor has two cluster lists: “input” and “output”. Server clusters reside on the input list, and client clusters reside on the output list. These simple descriptors are discoverable by other devices in the networks and are used to steer clients to the correct servers. For example, they to allow light switches to find lights.

2.4 Cluster pointer

All clusters are represented by the same `ZbZclClusterT` datatype, which represents an instance of a specific cluster. In general, the internal structure of this datatype is of no use to the application; however, it is very important because it is used throughout the cluster APIs as the generic representation of any cluster, regardless of the type of cluster. Because of this, many APIs are generic and work with any cluster. For example `ZbZclReadReq()` allows the attribute to be read in any cluster, whereas ZbZclDoorLockClientLockReq only works with a Door Lock client cluster handle, but both use the same `ZbZclClusterT` datatype. The distinction between different functions is made contextually.

2.5 Allocation functions

All clusters include server and client allocation functions, taking the form of:

```
app->x_client_cluster = ZbZclxClientAlloc(zb, endpoint, …)
```

or:

```
app->x_server_cluster = ZbZclxServerAlloc(zb, endpoint, …) if (app->x_server_cluster ==
NULL) {
```

The allocation functions return NULL on error, otherwise a cluster handle is returned. In general, the application never examines the contents of this struct. Instead this handle is used in most cluster library applications.

Like most ZSDK API functions, the allocation functions take the structure `ZigbeeT *` stack pointer as their first argument. This binds the new cluster instance to the stack instance. After which all ZCL API functions take the structure `ZbZclClusterT *` returned by the allocation function, the reference to the newly created cluster instance.

The second argument for allocation functions is the endpoint ID. This binds the newly created cluster to the endpoint given as the argument. Multiple cluster instances can be bound to the same endpoint, provided there is only one instance of any given cluster. The remainder of the arguments in an allocation function are specific to the cluster.

Server clusters with multiple commands usually take a structure with multiple callbacks, one for each command that the application supports. The application provides a callback for each command that it supports. If the application provides NULL for the entire structure pointer or a specific command callback, the cluster responds with a ZCL_STATUS_UNSUPP_CLUSTER_COMMAND default response for the specific command (or every command if the entire structure pointer is NULL).

Here is an example of how an application implements such a callback, providing the `app_get_profile_info()` function is called whenever the ZCL_ELEC_MEAS_CLI_GET_PROFILE_INFO command is received.

```
enum ZclStatusCodeT app_get_profile_info(struct ZbZclClusterT *cluster, struct
ZbZclAddrInfoT *src_info, void *arg)
{
    return ZCL_STATUS_SUCCESS;
}…
```

```
struct ElecMeasSvrCallbacksT callbacks = { app_get_profile_info,NULL,};
…
```

```
cluster = ZbZclElecMeasServerAlloc(zb, endpoint, callbacks, app);…
```

Note: *In this example, the Get Measurement Profile callback is declared NULL. When this command is received, a ZCL_STATUS_UNSUPP_CLUSTER_COMMAND default response is automatically sent. The `enum ZclStatusCodeT` defines the available status codes.*

2.6 Command request function

The following sample code is the function used to send ZCL commands:

```
enum ZclStatusCodeT ZbZclCommandReq(struct ZigBeeT *zb, struct ZbZclCommandReqT*zclReq, void
(*callback)(struct ZbZclCommandRspT *rsp, void *arg), void *arg);
```

2.6.1 Command request function parameters

Table 2. **Command request function parameters**

Name	Description
`zb`	Zigbee® stack instance
`zclReq`	ZCL request structure, detailed in the next section
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

2.7 Command request structure

The following sample code is the ZCL command request structure:

```
typedef struct ZbZclCommandReqT
{
   struct ZbApsAddrT dst; uint16_t profileId;
   enum ZbZclClusterIdT clusterId; uint16_t srcEndpt;
   uint16_t txOptions;
   bool discoverRoute;
   uint8_t radius;
   struct ZbZclHeaderT hdr;
   const void *payload; unsigned int length;
}
ZbZclCommandReqT;
```

This structure is mapped to the structure as defined in ZCL:

Table 3. **Command request structure parameters**

Exegin	ZCL 7	Description
`dst`	`DstAddrMode, DstAddress, DstEndpoint`	APS address information data structure
`profileId`	`ProfileId`	-
`clusterId`	`ClusterId`	-
`srcEndpt`	`SrcEndpoint`	-
`txOptions`	`TxOptions`	For example: ZB_APSDE_DATAREQ_TXOPTION S_ACK
`discoverRoute`	-	Used for discovery if the destination is unknown. If a route discovery is performed separately using `ZbNlmeRouteDiscWait()`, then the `discoverRoute` can be set to zero, decreasing the length of time an APS data request may take if there is a problem sending the packet to the target.
`radius`	`Radius`	-
`hdr`	Frame control, manufacturer code, transaction sequence number, command identifier	Payload
`Frame payload`	if `txOptions & ZB_APSDE_DATAREQ_TXOPTION S_VECTO`, payload is a pointer to list of struct `ZbApsBufT`, and length is the number of struct `ZbApsBufTitems` in the list	Length

2.8 ZCL callbacks

The following sections describe the callback used by ZCL commands. The callback function is called when the associated ZCL response is received, or if there is an error.

2.8.1 Function

```
(*callback)(struct ZbZclCommandRspT *rsp, void *arg)
```

2.8.2 ZCL callbacks parameters

Table 4. **ZCL callbacks paramters**

Name	Description
`rsp`	ZCL command response structure
`arg`	Pointer to application data provided in initiating API call

2.8.3 ZCL command response structure

The following code sample in the command response structure used by ZCL Callbacks.

```
struct ZbZclCommandRspT
{
   enum ZbStatusCodeT aps_status;
   enum ZclStatusCodeT status;
   struct ZbApsAddrT src;
   uint16_t profileId;
   enum ZbZclClusterIdT clusterId;
   uint8_t linkQuality;
   struct ZbZclHeaderT hdr;
   const uint8_t *payload; uint16_t length;
}
```

Table 5. **ZCL command response structure parameter**

Exegin	ZCL 7	Description
`aps_status`	-	APSDE-DATA.confirm status
`status`	`Status`	Status of the ZCL response. If a default response, it is the status code found within the response. If a cluster-specific response, it is set to ZB_STATUS_SUCCESS, and the application must parse the payload, if any, for any embedded status
`src`	`SrcEndpoint`	-
`profileId`	`ProfileId`	-
`clusterId`	`ClusterId`	-
`linkQuality`	`LinkQuality`	-
`hdr`	Frame control, manufacturer code, transaction sequence number, command identifier	-
`payload`	Frame payload	-
`length`	`ASDULength`	-

3 Addressing

The destination of ZCL messages is specified using the APS `ZbApsAddrT` structure. This structure provides the destination addressing, and either the short or extended address of the target node (the source addressing comes from the originating cluster instance). It is a burden for the application to keep track of the destination of each message.

To assist in this process the APS layer provides the binding mechanism.

Instead of tracking the clients and addressing them individually each and every time a message needs to be sent, the application sets up bindings. Bindings are addressing information stored in the binding table. When it is time to send a message the application specifies that the addressing is to binding, and the address information is automatically added from the binding table. This is done by specifying ZB_APSDE_ADDRMODE_NOTPRESENT as the mode in `ZbApsAddrT`. If the application wants to configure the destination to use bindings, use the `ZbApsAddrBinding` special global structure to copy this configuration from or use as a pointer reference.

Prior to using the `ZbApsAddrBinding`, bindings must be configured in the binding table using the `ZbApsmeBindReq()`. The bind request specifies a single `ClusterId` for the binding; For this `ClusterId` the binding then associates a source address and endpoint with a destination address and endpoint. Normally the source address is the device itself. When a sender has multiple cluster instances, they reside on separate endpoints; to use bindings with that endpoint, there must be bindings for each source endpoint. Each binding specifies a destination address and endpoint. A single source endpoint may also have a binding to multiple destination endpoints and even multiple endpoints on multiple address; it all depends on the bindings that are configured in the binding table.

One thing to note is that when using binding for the addressing it is assumed that at least one suitable binding exists. If no binding exists, a status of ZB_APS_STATUS_INVALID_BINDING is returned. However, if this is acceptable for the application, it may ignore this status.

The binding mechanism is a general purpose APS layer mechanism available for any APS layer purpose. So, a cluster needs to send ZCL reports where bindings are used (reports are sent to the available binding). Additionally, the Poll Control server cluster relies on bindings to send check-in requests; it sends "check in" requests to every client for which there is a binding present in its local binding table.

Bindings may also be configured from a remote node using the `ZbZdoMgmtBindReq()`. This is useful in cases such as reporting or the Poll Control clients which need to establish bindings on the remote node back to themselves.

In addition to manually establishing a binding, locally through `ZbApsmeBindReq()` or remotely through `ZbZdoMgmtBindReq()` there is the Finding and Binding mechanism. When initiated, Finding and Binding scans the local endpoints for client clusters, then locates remote server endpoints using Zigbee® service discovery mechanisms, and establishes bindings on the client to discovered server(s). Finding and Binding is triggered automatically by setting the ZB_BDB_CommissioningMode to BDB_COMMISSION_MODE_FIND_BIND prior to startup. When triggered automatically, the Finding and Binding procedure starts approximately 5 seconds after a node joins a network. Additionally, Finding and Binding is started manually from the application on all endpoints by calling `ZbStartupFindBindStart()`, or calling `ZbStartupFindBindStartEndpoint()` to start only from a particular endpoint.

4 Special clusters

Some clusters require the application developer to have a better understanding of both their function and their interactions with other clusters in order for the cluster to be properly implemented. Additional information concerning three such clusters is contained within this section.

Those Clusters are:

- Scenes cluster
- Alarms cluster
- CBKE Cluster

4.1 Scenes cluster

A scene is a set of values for attributes from multiple clusters capable of being applied at the same time. The few clusters that support scenes are identified by a section with the title "Scene Table Extensions" in the ZCL 8 Specification [1] section for those cluster. There is only one scene table (list of attributes) for a cluster that supports scenes, and when a scene is invoked all scene table attributes are set to the values given in the scene table.

To use the scene table for a cluster, the cluster must reside on an endpoint which also hosts an instance of the Scenes cluster. There may be multiple scene table supporting clusters on a given endpoint. A scene is defined in the scenes cluster and contains scene tables for one or more clusters. Through the use of group addressing a scene may be applied to multiple endpoints on a node.

A scene may be created by the Scene cluster using the Add Scene command, where the application manually defines the scene table for each cluster included in that scene. All attributes must have values in the scene table, but inclusion of individual clusters is optional. A scene may also be created using the Store Scene command where the current value of all the attributes in the cluster at the time the Store Scene command is issued are recorded in the scene table for later use.

The Scenes cluster Recall Scene command takes the scene table for each cluster in that scene and sets the values of every scene table attribute.

For example, a node could contain three endpoints:

- `0x01` with the On/Off cluster and Window covering cluster
- `0x02` with the On/Off cluster and Door lock cluster
- `0x03` with the On/Off cluster and Level cluster.

A scene is defined with a scene tables for the:

- On/Off cluster: `OnOff = On`
- Level cluster: `CurrentLevel = 50%`
- Door lock cluster: `LockState = Locked`

Additionally:

- Endpoints `0x01` and `0x02` are in group `0x0001`
- Endpoint `0x03` is not in group `0x0001`

If the Scenes cluster Recall Scenes command is issued with group address 0x0001 and the scene defined above, then on endpoint `0x01` and `0x02` the On/Off cluster OnOff attribute is set on and the DoorLock on endpoint `0x02` is be locked.

The Window covering cluster on endpoint `0x01` is not be affected because this scene does not include a scene table for this cluster and all of endpoint `0x03` is be unaffected because it is not in group 0x0001.

For more information about the Scenes cluster, see Section 3.7 in [1].

4.2 Alarms cluster

Zigbee® defines an alarm as the occurrence of a specific condition. Individual clusters (such as Basic cluster, Power configuration cluster, Door lock cluster, Ballast configuration cluster, and so on) define these conditions and a corresponding alarm code.

For the definition of the alarm condition its corresponding code for a specific cluster, see [1]

Alarm conditions are typically defined in terms of a cluster attribute. For example, the Power configuration cluster defines alarm code 0x00 for the alarm generated when the `MainsVoltage` attribute drops below the value specified in the `MainsVoltageMinThreshold` attribute for a period greater than the `MainsVoltageDwellTripPoint` attribute in seconds.

Clusters typically have an additional AlarmMask attribute which is a bitmask that allows the client to enable or disable the generation of individual alarms when the corresponding alarm condition is met.

It is the responsibility of the cluster application implementation to detect the alarm condition, check the alarm mask, and when needed initiate generation of the alarm by calling `ZbZclClusterSendAlarm()` (defined in `zcl.h`), resulting in the sending of an alarm command. It is important to note that this alarm command is not sent from the originating cluster. Instead, it is sent from an instance of the Alarms cluster that must reside on the same endpoint as the originating cluster.

The Alarms cluster sends the alarm command to all clients with bindings to the alarm cluster on this endpoint. It also adds alarm details to an internal log. The alarm cluster provides commands that allow clients to query this alarm log. The alarm log is shared by all clusters on the same endpoint. To receive alarms, clients must bind to the alarms cluster on the same endpoint as the alarm generating cluster. For clusters that support an alarm mask, any client enables and disables the generation of alarm commands by setting and clearing the mask bit in the originating cluster. The mask controls sending of alarms to all bound clients.

Some alarm conditions do not automatically reset and must be manually reset by the client. The Alarms cluster provides the Reset Alarm and Reset all Alarms commands for this reason the application should register a callback for each endpoint with cluster(s) that require resetting using `ZbZclClusterRegisterAlarmResetHandler()`. The callback handles the specific cluster(s) against which it was registered.

When the Alarms cluster receives a Reset Alarm or Reset all Alarms command, the cluster application callback is invoked and then handles whatever is necessary to internally reset to be able to detect new occurrences of the alarm condition. The same callback is invoked for both commands. When the Reset all Alarms command is received the callback is invoked with a `0xFF` Alarm Code and `0xFFFF` Cluster ID. When a callback is provided, the function return code is provided as the status in the default response. When no callbacks are provided the alarm cluster sends a default response with a status of SUCCESS.

As a summary:

- When an endpoint contains a cluster that generates alarms, it is the application responsibility to instantiate the alarms cluster on that endpoint.
- It is the responsibility of the cluster implementation to:
  - Detect any alarm conditions
  - Check the alarm mask (where supported)
  - Generate an alarm by calling `ZbZclClusterSendAlarm()`.
- If the alarm conditions for any cluster on an endpoint needs to be manually reset, then the application must register a callback for each endpoint with the clusters that require resetting using `ZbZclClusterRegisterAlarmResetHandler()`.

For more information about the Alarms cluster, see Section 3.11 in [1].

4.3 CBKE Cluster

The certificate-based key establishment (CBKE) cluster is handled internally by the stack. When CBKE is enabled, the stack creates an instance of the CBKE cluster. There is no exposed public API to the CBKE cluster itself. CBKE is configured in the `security.cbke` (`struct ZbStartupCbkeT`) section of the `ZbStartupT` startup config. The basic setup procedure is to enable the supported suites in `suite_mask` and load the corresponding suite configuration and certificates for the enabled suites into the startup config before starting the stack.

4.3.1 ZbStartupCbkeT

CBKE configuration struct for `ZbStartup`. This is only applicable if the `suite_mask` is non-zero.

Table 6. **ZbStartupCbkeT Parameters**

Parameter	Description
`uint8_t endpoint`	Endpoint to assign ZCL Key Exchange cluster. Default is ZB_ENDPOINT_CBKE_DEFAULT (240)
`uint16_t deviceId`	Device Id to assign to the endpoint created for the ZCL Key Exchange cluster. Default is ZCL_DEVICE_METER.
`uint16_t suite_mask`	The Key Exchange suite bitmask. E.g. ZCL_KEY_SUITE_CBKE2_ECMQV for CBKE version 2 (cbke_v2).
`struct ZbZclCbkeInfoT cbke_v1`	CBKE version 1 certificate and security keys configuration. Only applicable if ZCL_KEY_SUITE_CBKE_ECMQV is set in `suite_mask`.
`struct ZbZclCbke2InfoT cbke_v2`	CBKE version 2 certificate and security keys configuration. Only applicable if ZCL_KEY_SUITE_CBKE2_ECMQV is set in `suite_mask`.
`bool tc_keepalive_server_enable`	Trust Center Keep Alive Server 'Jitter' attribute value in seconds. If zero, let the stack choose a default value.
`void (*tcso_callback)(enum ZbTcsoStatusT status, void *arg)`	Application callback which is called for any Trust Center Swap Out (TCSO) process initiated by the Keep Alive Client cluster. This includes when a TCSO has been started and the final outcome of a TCSO process.
`void *tcso_arg`	Application callback argument for `tcso_callback`.

5 Cluster API Documentation

5.1 Prototype_key

Table 7. List of all the prototypes

	Function	Definition
Alarms cluster	ZbZclAlarmClientAlloc	Create a new instance of the Alarms Client cluster.
	ZbZclAlarmClientGetAlarmReq	Send a Get Alarm command.
	ZbZclAlarmClientResetAlarmLogReq	Send a Reset Alarm Log command.
	ZbZclAlarmClientResetAlarmReq	Send a Reset Alarm command
	ZbZclAlarmClientResetAllAlarmsReq	Send a Reset All Alarms command
	ZbZclAlarmServerAlloc	Create a new instance of the Alarms Server cluster.
	ZbZclAlarmsAttrT	ZbZclAlarmsAttrT
Ballast configuration cluster	ZbZclBallastConfigClientAlloc	Create a new instance of the ballast configuration client cluster
	ZbZclBallastConfigServerAlloc	Create a new instance of the Ballast Configuration Server cluster
	ZbZclBallastConfigSvrAttrT	-
Basic cluster	Section 5.4.2.1 ZbZclBasicClientAlloc	Create a new instance of the Alarms Client cluster.
	Section 5.4.2.2 ZbZclBasicClientResetReq	Send a Reset to Factory Defaults command
	Section 5.4.3.1 ZbZclBasicSvrAttrT	-
Color control cluster	ZbZclColorClientAlloc	Instantiate a new instance of the Color Control client cluster
	ZbZclColorClientColorLoopSetReq	Send a Color Loop Set command
	ZbZclColorClientMoveColorTempReq	Send a Move Color Temperature command
	ZbZclColorClientMoveColorXYReq	Send a Move Color command
	ZbZclColorClientMoveHueEnhReq	Send an Enhanced Move Hue command
	ZbZclColorClientMoveHueReq	Send a Move Hue command
	ZbZclColorClientMoveSatReq	Send a Move Saturation command
	ZbZclColorClientMoveToColorTempReq	Send a Move to Color Temperature command
	ZbZclColorClientMoveToColorXYReq	Send a Move to Color command
	ZbZclColorClientMoveToHueEnhReq	Send an Enhanced Move to Hue command
	ZbZclColorClientStepColorXYReq	Send a Step Color command
	ZbZclColorClientMoveToHueSatEnhReq	Send an Enhanced Move to Hue and Saturation command
	ZbZclColorClientMoveToHueSatReq	Send a Move to Hue and Saturation command
	ZbZclColorClientMoveToSatReq	Send a Move to Saturation command
	ZbZclColorClientStepColorTempReq	Send a Step Color Temperature command
	ZbZclColorClientStepHueEnhReq	Send an Enhanced Step Hue command
	ZbZclColorClientStepHueEnhReq	Send an Enhanced Step Hue command
	ZbZclColorClientStepHueReq	Send a Step Hue command

	Function	Definition
Color control cluster	ZbZclColorClientStepSatReq	Send a Step Saturation command
	ZbZclColorClientStopMoveStepReq	Send a Stop Move Step command
	ZbZclColorServerAlloc	Instantiate a new instance of the Color Control server cluster
	ZbZclColorSvrAttrT	-
	ZbZclColorClientColorLoopSetReqT	Color Loop Set command structure
	ZbZclColorClientMoveColorTempReqT	Move Color Temperature command structure
	ZbZclColorClientMoveColorXYReqT	Move Color command structure
	ZbZclColorClientMoveHueEnhReqT	Enhanced Move Hue command structure
	ZbZclColorClientMoveHueReqT	Move Hue command structure
	ZbZclColorClientMoveSatReqT	Move Saturation command structure
	ZbZclColorClientMoveToColorTempReqT	Move to Color Temperature command structure
	ZbZclColorClientMoveToColorXYReqT	Move to Color command structure
	ZbZclColorClientMoveToHueEnhReqT	Enhanced Move to Hue command structure
	ZbZclColorClientMoveToHueReqT	Move to Hue command structure
	ZbZclColorClientMoveToHueSatEnhReqT	Enhanced Move to Hue and Saturation command structure
	ZbZclColorClientMoveToHueSatReqT	Move to Hue and Saturation command structure
	ZbZclColorClientMoveToSatReqT	Move to Saturation command structure
	ZbZclColorClientStepColorTempReqT	Step Color Temperature command structure
	ZbZclColorClientStepColorXYReqT	Step Color command structure
	ZbZclColorClientStepHueEnhReqT	Enhanced Step Hue command structure
	ZbZclColorClientStepHueReqT	Step Hue command structure
	ZbZclColorClientStepSatReqT	Step Saturation command structure
	ZbZclColorClientStopMoveStepReqT	Stop Move Step command structure
	ZbZclColorServerCallbacksT	Color Control Server callbacks configuration
Commissioning cluster	ZbZclCommissionClientAlloc	Create a new instance of the Commissioning Client cluster
	ZbZclCommissionClientEnable	Enable Commissioning Client by configuring MAC layer to listen for packets.
	ZbZclCommissionClientSendResetStartup	Send a Reset Startup Parameters command
	ZbZclCommissionClientSendRestart	Send a Restart Device command
	ZbZclCommissionClientSendRestoreStartup	Send a Restore Startup Parameters command
	ZbZclCommissionServerAlloc	Create a new instance of the Commissioning Server cluster
	ZbZclCommissionServerEnable	Enable the Commissioning Server by configuring the MAC layer to listen for packets. If enable is false, then Commissioning Server stops processing any received Commissioning packets.
	ZbZclCommissionServerGetStartup	Load startup configuration from Cluster Server's attributes to the stack's ZbStartupT structure
	ZbZclCommissionServerResetStartup	Reset startup configurations cluster attributes back to defaults
	ZbZclCommissionServerSendResetStartupRsp	Send a Reset Startup Parameters Response command

	Function	Definition
Commissioning cluster	ZbZclCommissionServerSendRestartRsp	Send a Restart Device Response command
	ZbZclCommissionServerSendRestoreStartupRsp	Send a Restore Startup Parameters Response command
	ZbZclCommissionServerSendSaveStartupRsp	Send a Save Startup Parameters Response command
	ZbZclCommissionServerAttrT	-
	ZbZclCommissionClientEnableInfoT	Commissioning Client Enable Information structure
	ZbZclCommissionClientResetStartup	Reset Startup Parameters command structure
	ZbZclCommissionClientRestartDev	Restart Device command structure
	ZbZclCommissionClientRestoreStartup	Restore Startup Parameters command structure
	ZbZclCommissionClientSaveStartup	Save Startup Parameters command structure
	ZbZclCommissionServerCallbacksT	Commissioning Server callbacks configuration
	ZbZclCommissionServerEnableInfoT	Commissioning Server Enable Information structure
	ZbZclCommissionServerResetStartupRsp	Reset Startup Parameters Response command structure
	ZbZclCommissionServerRestartDevRsp	Restart Device Response command structure
	ZbZclCommissionServerRestoreStartupRsp	Restore Startup Parameters Response command structure
Dehumidification Control cluster	ZbZclDehumCtrlClientAlloc	Instantiate a new instance of the Dehumidification Control client cluster
	ZbZclDehumCtrlServerAlloc	Instantiate a new instance of the Dehumidification Control server cluster
	ZbZclDehumCtrlServerAttrT	-
Device temperature configuration cluster	ZbZclDevTempClientAlloc	Create a new instance of the Device Temperature Configuration Client cluster
	ZbZclDevTempServerAlloc	Instantiate a new instance of the Device Temp client cluster.
	ZbZclDeviceTempAlarmCode	-
	ZbZclDeviceTempAlarmMask	-
	ZbZclDeviceTempSvrAttrT	-
Diagnostics cluster	ZbZclDiagClientAlloc	Create a new instance of the Diagnostics Client cluster
	ZbZclDiagServerAlloc	Create a new instance of the Diagnostics Server cluster. Only one Diagnostics Server is allocated on the device
	ZbZclDiagSvrAttrT	-
Door lock cluster	ZbZclDoorLockClientAlloc	Create a new instance of the Door Lock Client cluster
	ZbZclDoorLockClientClrAllPinReq	Send a Clear All PIN Codes request command
	ZbZclDoorLockClientClrAllRfidReq	Send a Clear All RFID Codes request command
	ZbZclDoorLockClientClrHDScheduleReq	Send a Clear Holiday Schedule request command
	ZbZclDoorLockClientClrPinReq	Send a Clear PIN Code request command
	ZbZclDoorLockClientClrRfidReq	Send a Clear RFID Code request command

	Function	Definition
Door lock cluster	ZbZclDoorLockClientClrWDScheduleReq	Send a Clear Weekday Schedule request command
	ZbZclDoorLockClientClrYDScheduleReq	Send a Clear Year Day Schedule request command
	ZbZclDoorLockClientGetLogReq	Send a Get Log Record request command
	ZbZclDoorLockClientGetPinReq	Send a Get PIN Code request command
	ZbZclDoorLockClientGetRfidReq	Send a Get RFID Code request command
	ZbZclDoorLockClientGetUserStatusReq	Send a Get User Status request command
	ZbZclDoorLockClientGetHDScheduleReq	Send a Get Holiday Schedule request command
	ZbZclDoorLockClientGetUserTypeReq	Send a Get User Type request command
	ZbZclDoorLockClientGetWDScheduleReq	Send a Get Weekday Schedule request command
	ZbZclDoorLockClientGetYDScheduleReq	Send a Get Year Day Schedule request command
	ZbZclDoorLockClientLockReq	Send a Lock Door request command
	ZbZclDoorLockClientSetHDScheduleReq	Send a Set Holiday Schedule request command
	ZbZclDoorLockClientSetPinReq	Send a Set PIN Code request command
	ZbZclDoorLockClientSetRfidReq	Send a Set RFID Code request command
	ZbZclDoorLockClientSetUserStatusReq	Send a Set User Status request command
	ZbZclDoorLockClientSetUserTypeReq	Send a Set User Type request command
	ZbZclDoorLockClientSetWDScheduleReq	Send a Set Weekday Schedule request command
	ZbZclDoorLockClientSetYDScheduleReq	Send a Set Year Day Schedule request command
	ZbZclDoorLockClientToggleReq	Send Toggle request command
	ZbZclDoorLockClientUnlockReq	Send an Unlock Door request command
	ZbZclDoorLockClientUnlockTimeoutReq	Send an Unlock with Timeout request command
	ZbZclDoorLockServerAlloc	Create a new instance of the Door Lock Server cluster
	ZbZclDoorLockServerSendClrAllPinRsp	Send a Clear All PIN Codes response command
	ZbZclDoorLockServerSendClrAllRfidRsp	Send a Clear All RFID Codes response command
	ZbZclDoorLockServerSendClrHDScheduleRsp	Send a Clear Holiday Schedule response command
	ZbZclDoorLockServerSendClrPinRsp	Send a Clear PIN Code response command
	ZbZclDoorLockServerSendClrRfidRsp	Send a Clear RFID Code response command
	ZbZclDoorLockServerSendClrWDScheduleRsp	Send a Clear Weekday Schedule response command
	ZbZclDoorLockServerSendClrYDScheduleRsp	Send a Clear Year Day Schedule response command
	ZbZclDoorLockServerSendGetHDScheduleRsp	Send a Get Holiday Schedule response command
	ZbZclDoorLockServerSendGetLogRsp	Send a Get Log Record response command
	ZbZclDoorLockServerSendGetPinRsp	Send a Get PIN Code response command
	ZbZclDoorLockServerSendGetRfidRsp	Send a Get RFID Code response command
	ZbZclDoorLockServerSendGetUserStatusRsp	Send a Get User Status response command

	Function	Definition
Door lock cluster	ZbZclDoorLockServerSendGetUserTypeRsp	Send a Get User Type response command
	ZbZclDoorLockServerSendGetWDScheduleRsp	Send a Get Weekday Schedule response command
	ZbZclDoorLockServerSendGetYDScheduleRsp	Send a Get Year Day Schedule response command
	ZbZclDoorLockServerSendLockRsp	Send a Lock Door response command
	ZbZclDoorLockServerSendSetHDScheduleRsp	Send a Set Holiday Schedule response command
	ZbZclDoorLockServerSendSetPinRsp	Send a Set PIN Code response command
	ZbZclDoorLockServerSendSetRfidRsp	Send a Set RFID Code response command
	ZbZclDoorLockServerSendSetUserStatusRsp	Send a Set User Status response command
	ZbZclDoorLockServerSendSetUserTypeRsp	Send a Set User Type response command
	ZbZclDoorLockServerSendSetWDScheduleRsp	Send a Set Weekday Schedule response command
	ZbZclDoorLockServerSendSetYDScheduleRsp	Send a Set Year Day Schedule response command
	ZbZclDoorLockServerSendToggleRsp	Send a Toggle response command
	ZbZclDoorLockServerSendUnlockRsp	Send an Unlock Door response command
	ZbZclDoorLockServerSendUnlockTimeoutRsp	Send an Unlock with Timeout response command
	ZbZclDoorLockClrAllPinRspT	Clear All PIN Codes response structure
	ZbZclDoorLockClrAllRfidRspT	Clear All RFID Codes response structure
	ZbZclDoorLockClrHDScheduleReqT	Clear Holiday Schedule request structure
	ZbZclDoorLockClrHDScheduleRspT	Clear Holiday Schedule response structure
	ZbZclDoorLockClrPinReqT	Clear PIN Code request structure
	ZbZclDoorLockClrPinRspT	Clear PIN Code response structure
	ZbZclDoorLockClrRfidReqT	Clear RFID Code request structure
	ZbZclDoorLockClrRfidRspT	Clear RFID Code response structure
	ZbZclDoorLockClrWDScheduleReqT	Clear Weekday Schedule request structure
	ZbZclDoorLockClrWDScheduleRspT	Clear Weekday Schedule response structure
	ZbZclDoorLockClrYDScheduleReqT	Clear Year Day Schedule request structure
	ZbZclDoorLockClrYDScheduleRspT	Clear Year Day Schedule response structure
	ZbZclDoorLockGetHDScheduleReqT	Get Holiday Schedule request structure
	ZbZclDoorLockGetHDScheduleRspT	Get Holiday Schedule response structure
	ZbZclDoorLockGetLogReqT	Get Log Record request structure
	ZbZclDoorLockGetLogRspT	Get Log Record response structure
	ZbZclDoorLockGetPinReqT	Get PIN Code request structure
	ZbZclDoorLockGetPinRspT	Get PIN Code response structure
	ZbZclDoorLockGetRfidReqT	Get RFID Code request structure
	ZbZclDoorLockGetRfidRspT	Get RFID Code response structure
	ZbZclDoorLockGetUserStatusReqT	Get User Status request structure
	ZbZclDoorLockGetUserStatusRspT	Get User Status response structure
	ZbZclDoorLockGetUserTypeReqT	Get User Type request structure
	ZbZclDoorLockGetUserTypeRspT	Get User Type response structure

	Function	Definition
Door lock cluster	ZbZclDoorLockGetWDScheduleReqT	Get Weekday Schedule request structure
	ZbZclDoorLockGetWDScheduleRspT	Get Weekday Schedule response structure
	ZbZclDoorLockGetYDScheduleReqT	Get Year Day Schedule request structure
	ZbZclDoorLockGetYDScheduleRspT	Get Year Day Schedule response structure
	ZbZclDoorLockLockDoorReqT	Lock Door request structure
	ZbZclDoorLockLockDoorRspT	Lock Door response structure
	ZbZclDoorLockSetHDScheduleReqT	Set Holiday Schedule request structure
	ZbZclDoorLockSetHDScheduleRspT	Set Holiday Schedule response structure
	ZbZclDoorLockSetPinReqT	Set PIN Code request structure
	ZbZclDoorLockSetPinRspT	Set PIN Code response structure
	ZbZclDoorLockSetRfidReqT	Set RFID Code request structure
	ZbZclDoorLockSetRfidRspT	Set RFID Code response structure
	ZbZclDoorLockSetUserStatusReqT	Set User Status request structure
	Section 5.10.3.39 ZbZclDoorLockSetUserStatusRspT	Set User Status response structure
	ZbZclDoorLockSetUserTypeReqT	Set User Type request structure
	ZbZclDoorLockSetWDScheduleRspT	Set Weekday Schedule response structure
	ZbZclDoorLockSetYDScheduleReqT	Set Year Day Schedule request structure
	ZbZclDoorLockSetYDScheduleRspT	Set Year Day Schedule response structure
	ZbZclDoorLockToggleReqT	Toggle request structure
	ZbZclDoorLockToggleRspT	Toggle response structure
	ZbZclDoorLockUnlockDoorReqT	Unlock Door request structure
	ZbZclDoorLockUnlockDoorRspT	Unlock Door response structure
	ZbZclDoorLockUnlockTimeoutReqT	Unlock with Timeout request structure
	ZbZclDoorLockUnlockTimeoutRspT	Unlock with Timeout response structure
Demand response and load control cluster	ZbZclDrlcClientAlloc	Create a new instance of the DRLC Client cluster
	ZbZclDrlcClientCommandGetEventsReq	Send a Get Scheduled Events
	ZbZclDrlcClientCommandReportStatusReq	Send a Report Event Status Command
	ZbZclDrlcClientGetEventList	Send a Get Event List command
	ZbZclDrlcServerAlloc	Create a new instance of the DRLC Server cluster
	ZbZclDrlcServerCommandCancelAllReq	Send a Cancel All Load Control Event command
	ZbZclDrlcServerCommandCancelReq	Send a Cancel Load Control Event command
	ZbZclDrlcServerCommandEventReq	Send a Load Control Event command
	ZbZclDrlcCliAttrT	DRLC client attribute IDs
	ZbZclDrlcCancelT	Cancel Load Control Event command structure
	ZbZclDrlcClientCallbacksT	DRLC Client callbacks configuration
	ZbZclDrlcEventT	Load Control Event command structure
	ZbZclDrlcGetEventsReqT	Get Scheduled Events command structure
	ZbZclDrlcServerCallbacksT	DRLC Server callbacks configuration
	ZbZclDrlcStatusT	Report Event Status command structure

	Function	Definition
Electrical measurement cluster	ZbZclElecMeasClientAlloc	Create a new instance of the Electrical Measurement Client cluster
	ZbZclElecMeasClientGetMeasProfileReq	Send a Get Measurement Profile command
	ZbZclElecMeasClientGetProfileInfoReq	Send a Get Profile Info command
	ZbZclElecMeasServerAlloc	Create a new instance of the Electrical Measurement Server cluster
	ZbZclElecMeasServerSendMeasProfileRsp	Send a Get Measurement Profile response
	ZbZclElecMeasServerSendProfileInfoRsp	Send a Get Profile Info response
	ZbZclElecMeasSvrAttrT	Electrical Measurement server attribute IDs
	ZbZclElecMeasClientGetMeasProfileReqT	Get Measurement Profile command structure
	ZbZclElecMeasSvrCallbacksT	Electrical Measurement Server callbacks configuration
	ZbZclElecMeasSvrGetMeasProfileRspT	Get Measurement Profile response structure
	ZbZclElecMeasSvrGetProfileInfoRspT	Get Profile Info response structure
Fan control cluster	ZbZclFanClientAlloc	Create a new instance of the Fan Control Client cluster
	ZbZclFanServerAlloc	Create a new instance of the Fan Control Server cluster
Groups cluster	ZbZclGroupsClientAddIdentifyingReq	Send an Add Group If Identifying command
	ZbZclGroupsClientAddReq	Send an Add Group command
	ZbZclGroupsClientAlloc	Create a new instance of the Groups Client cluster
	ZbZclGroupsClientGetMembershipReq	Send a Get Group Membership command
	ZbZclGroupsClientRemoveAllReq	Send a Remove All Groups command
	ZbZclGroupsClientRemoveReq	Send a Remove Group command
	ZbZclGroupsClientViewReq	Send a View Group command
	ZbZclGroupsServerAlloc	Create a new instance of the Groups Server cluster
	ZbZclGroupsSvrAttrT	-
	ZbZclGroupsClientAddIdentifyingReqT	Add Group If Identifying command structure
	ZbZclGroupsClientAddReqT	Add Group command structure
	ZbZclGroupsClientGetMembershipReqT	Get Group Membership command structure
	ZbZclGroupsClientRemoveReqT	Remove Group command structure
	ZbZclGroupsClientViewReqT	View Group command structure
IAS ACE cluster	ZbZclIasAceClientAlloc	Allocate the IAS ACE Client cluster
	ZbZclIasAceClientCommandArmReq	Send an Arm command
	ZbZclIasAceClientCommandBypassReq	Send a Bypass command
	ZbZclIasAceClientCommandEmergencyReq	Send an Emergency command
	ZbZclIasAceClientCommandFireReq	Send a Fire command
	ZbZclIasAceClientCommandGetBypassedZoneListReq	Send a Get Bypassed Zone List command
	ZbZclIasAceClientCommandGetPanelStatusReq	Send a Get Panel Status changed request
	ZbZclIasAceClientCommandGetZoneIdMapReq	Send a Get Zone ID Map command
	ZbZclIasAceClientCommandGetZoneInfoReq	Send a Get Zone Info command

	Function	Definition
IAS ACE cluster	ZbZclIasAceClientCommandGetZoneStatusReq	Send a Get Zone Status command
	ZbZclIasAceClientCommandPanicReq	Send a Panic command
	ZbZclIasAceClientParseArmRsp	Parse an Arm Response command payload into a data structure
	ZbZclIasAceClientParseBypassRsp	Parse a Bypass Response command payload into a data structure
	ZbZclIasAceClientParseGetPanelStatusRsp	Parse a Get Panel Status Response command payload into a data structure
	ZbZclIasAceClientParseGetZoneIdMapRsp	Parse a Get Zone ID Map Response command payload into a data structure
	ZbZclIasAceClientParseGetZoneInfoRsp	Parse a Get Zone Info Response command payload into a data structure
	ZbZclIasAceClientParseGetZoneStatusRsp	Parse a Get Zone Status Response command payload into a data structure
	ZbZclIasAceClientParseSetBypassedZoneList	Parse a Set Bypassed Zone List Response command payload into a data structure
	ZbZclIasAceClientParseZoneStatusChanged	Parse a Zone Status Changed Response command payload into a data structure
	ZbZclIasAceServerAlloc	Create a new instance of the IAS ACE Server cluster
	ZbZclIasAceServerGetFreeZoneId	Returns the first free Zone ID not already in the Zone Table
	ZbZclIasAceServerPanelCodeConfig	Change the Panel Arm/Disarm Code
	ZbZclIasAceServerPanelStatusConfig	Update the Panel Status
	ZbZclIasAceServerZoneBypassConfig	Bypass zone if allowed
	ZbZclIasAceServerZoneBypassPerms	Configure Bypass Permissions
	ZbZclIasAceServerZoneStatusConfig	Configure Zone Status
	ZbZclIasAceServerZoneTableAdd	Add new zone entry
	ZbZclIasAceServerZoneTableAddrLookup	Returns address of paired zone, or 0 if not found
	ZbZclIasAceServerZoneTableDeleteByAddr	Delete a zone by address
	ZbZclIasAceServerZoneTableDeleteById	Delete a zone by zone ID
	ZbZclIasAceServerZoneTableIdLookup	Attempts to find a zone based on extended address, and returns the zone Id if found
	ZbZclIasAceClientCommandArmT	Arm command structure
	ZbZclIasAceClientCommandBypassT	Bypass command structure
	ZbZclIasAceClientCommandGetZoneInfoT	Get Zone Info command structure
	ZbZclIasAceClientCommandGetZoneStatusT	Get Zone Status command structure
	ZbZclIasAceServerCallbacksT	IAS ACE Server callbacks configuration
	ZbZclIasAceServerCommandArmRspT	Arm response structure
	ZbZclIasAceServerCommandBypassRspT	Bypass Response response structure
	ZbZclIasAceServerCommandGetPanelStatusRspT	Get Panel Status response structure
	ZbZclIasAceServerCommandGetZoneIdMapRspT	Get Zone ID Map response structure
	ZbZclIasAceServerCommandGetZoneInfoRspT	Get Zone Info response structure
	ZbZclIasAceServerCommandGetZoneStatusRspT	Get Zone Status response structure

	Function	Definition
IAS ACE cluster	ZbZclIasAceServerCommandSetBypassedZoneListT	Set Bypassed Zone List command structure
	ZbZclIasAceServerCommandZoneStatusChangedT	Zone Status Changed command structure
	ZbZclIasAceServerZoneTableAddT	Zone Table Add request structure
IAS WD cluster	ZbZclIasWdClientAlloc	Create a new instance of the IAS Warning Device Client cluster
	ZbZclIasWdServerAlloc	Create a new instance of the IAS Warning Device Server cluster
	ZbZclIasWdLevelT	IAS Warning Device Siren Level Field Values
	ZbZclIasWdSquawkModeT	IAS Warning Device Squawk Mode Field
	ZbZclIasWdStrobeT	IAS Warning Device Strobe Field
	ZbZclIasWdSvrAttrT	IAS Warning Device server attribute IDs
	ZbZclIasWdWarningModeT	IAS Warning Device Warning Modes
	ZbZclIasWdClientSquawkReqT	IAS Warning Device Client Squawk command structure
	ZbZclIasWdClientStartWarningReqT	IAS Warning Device Client Start Warning command structure
	ZbZclIasWdServerCallbacksT	IAS Warning Server callbacks configuration
IAS Zone cluster	ZbZclIasZoneClientAlloc	Create a new instance of the IAS Zone Client cluster
	ZbZclIasZoneClientInitiateAutoEnroll	Send a Zone Auto-Enroll request command
	ZbZclIasZoneClientInitiateNormalMode	Send a Initiate Normal Operation Mode request command
	ZbZclIasZoneClientInitiateTestMode	Send a Initiate Test Mode request command
	ZbZclIasZoneClientSendAutoEnrollResponse	Send a Zone Auto-Enroll response command
	ZbZclIasZoneServerAlloc	Create a new instance of the IAS Zone Server cluster
	ZbZclIasZoneServerEnrollRequest	Send a Zone Enroll request command Used with 'trip-to-pair'. Before sending a Zone Enroll Request, the IAS CIE must write to the IAS_CIE_Address attribute with its IEEE address.
	ZbZclIasZoneClientResponseCodeT	IAS Zone Enroll Response Code
	ZbZclIasZoneServerAttrT	IAS Zone server attribute IDs
	ZbZclIasZoneServerModeT	IAS Zone ZoneStatus Attribute Bit Test Value
	ZbZclIasZoneServerZoneStateT	IAS Zone ZoneState Attribute
	ZbZclIasZoneServerZoneStatusT	-
	ZbZclIasZoneServerZoneTypeT	-
	ZbZclIasZoneClientCallbacksT	IAS Zone Client callbacks configuration
	ZbZclIasZoneClientEnrollResponseT	Zone Enroll response structure
	ZbZclIasZoneClientTestModeReqT	Initiate Test Mode request structure
	ZbZclIasZoneServerCallbacksT	IAS Zone Server callbacks configuration
	ZbZclIasZoneServerEnrollRequestT	Zone Enroll request structure
	ZbZclIasZoneServerStatusChangeNotifyT	Zone State Change Notification request structure
Identify cluster	ZbZclIdentifyClientAlloc	Create a new instance of the Identify Client cluster

	Function	Definition
Identify cluster	ZbZclIdentifyServerAlloc	Create a new instance of the Identify Server cluster
	ZbZclIdentifyServerGetTime	Get the local Identify Server time
	ZbZclIdentifyServerSetCallback	Set the callback in the cluster private structure.
	ZbZclIdentifyServerSetTime	Set the local Identify Server time
	zcl_identify_identify_request	Send an Identify command
	zcl_identify_query_request	Send an Identify Query command
Illuminance level sensing cluster	ZbZclIllumLevelClientAlloc	Create a new instance of the Illuminance Level Sensing Client cluster
	ZbZclIllumLevelServerAlloc	Create a new instance of the Illuminance Level Sensing Server cluster
	ZbZclIllumLevelSvrAttrT	-
Illuminance measurement cluster	ZbZclIllumMeasClientAlloc	Create a new instance of the Illuminance Measurement Client cluster
	ZbZclIllumMeasServerAlloc	Create a new instance of the Illuminance Measurement Server cluster
	ZbZclIllumMeasSvrAttrT	Illuminance Measurement server attribute IDs
Level cluster	ZbZclLevelClientAlloc	Create a new instance of the Level Client cluster
	ZbZclLevelClientMoveReq	Send a Move command
	ZbZclLevelClientMoveToLevelReq	Send a Move to Level command
	ZbZclLevelClientStepReq	Send a Step command
	ZbZclLevelClientStopReq	Send a Stop command
	ZbZclLevelServerAlloc	Create a new instance of the Level Server cluster
	ZbZclLevelSvrAttrT	-
	ZbZclLevelClientMoveReqT	Move command structure `ZbZclLevelClientMoveReqT`
	ZbZclLevelClientMoveToLevelReqT	Move To Level command structure
	ZbZclLevelClientStepReqT	Step command structure
	ZbZclLevelClientStopReqT	Stop command structure
	ZbZclLevelServerCallbacksT	Level Server callbacks configuration
Messaging	ZbZclMsgClientAlloc	Create a new instance of the Messaging Client cluster
	ZbZclMsgClientConfReq	Send a Message Confirmation command
	ZbZclMsgClientGetLastReq	Send a Get Last Message command
	ZbZclMsgClientGetMsgCancelReq	Send a Get Message Cancellation command
	ZbZclMsgServerAlloc	Create a new instance of the Messaging Server cluster
	ZbZclMsgServerCancelAllReq	Send a Cancel All Messages command
	ZbZclMsgServerCancelMessageReq	Send a Cancel Message command
	ZbZclMsgServerAlloc	Create a new instance of the Messaging Server cluster
	ZbZclMsgServerCancelAllReq	Send a Cancel All Messages command
	ZbZclMsgServerCancelMessageReq	Send a Cancel Message command
	ZbZclMsgServerDisplayMessageReq	Send a Display Message command
	ZbZclMsgServerDisplayProtectedMsgReq	Send a Display Protected Message command

	Function	Definition
Messaging	ZbZclMsgClientCallbacksT	Messaging Client callbacks configuration
	ZbZclMsgConfirmEnhT	Enhanced Message Confirmation command structure
	Section 5.22.3.3 ZbZclMsgConfirmT	Message Confirmation command structure
	ZbZclMsgGetMsgCancellationT	Get Message Cancellation command structure
	ZbZclMsgMessageCancelAllT	Cancel All Messages command structure
	ZbZclMsgMessageCancelT	Cancel Message command structure
	ZbZclMsgMessageConfT	Message Confirmation callback structure
	ZbZclMsgMessageT	Display Message/Display Protected Message command structure
	ZbZclMsgServerCallbacksT	Messaging Server callbacks configuration
Metering cluster	ZbZclMeterClientAlloc	Create a new instance of the Metering Client cluster
	ZbZclMeterClientCommandGetProfileReq	Send a Get Profile command
	ZbZclMeterClientCommandGetSampledDataReq	Send a Get Sampled Data command
	ZbZclMeterClientCommandLocalChangeSupplyReq	Send a Local Change Supply command
	ZbZclMeterFormSampledData	Convert an array of 24-bit integers to the Zigbee frame format
	ZbZclMeterGetProfileIntervalPeriod	Convert the profile interval period enumerated value to a time in seconds
	ZbZclMeterServerAlloc	Create a new instance of the Metering Server cluster After initialization, `ZbZclClusterSetCallbackArg` is called with 'arg' to configure the application callback argument
	ZbZclMeterServerMirrorAlloc	Create a new instance of the Metering Server Mirror cluster
	ZbZclMeterServerSendGetProfileRsp	Send a Get Profile Response command
	ZbZclMeterServerSendGetSampledDataRsp	Send a Get Sampled Data Response command
	ZbZclMeteringSvrAttrT	-
	ZbZclMeterClientCallbacksT	Metering Client callbacks configuration
	ZbZclMeterClientGetProfileReqT	Get Profile command structure
	ZbZclMeterClientGetSampledDataReqT	Get Sampled Data command structure
	ZbZclMeterClientLocalChangeSupplyReqT	Metering Local Change Supply command structure
	ZbZclMeterServerCallbacksT	Metering Server callbacks configuration
	ZbZclMeterServerGetProfileRspT	Get Profile Response command structure
	ZbZclMeterServerGetSampledDataRspT	Get Sampled Data Response command structure
Meter identification cluster	ZbZclMeterIdClientAlloc	Instantiate a new instance of the Meter Identification Client cluster
	ZbZclMeterIdServerAlloc	Instantiate a new instance of the Meter Identification Server cluster
	ZbZclMeterIdSvrAttrT	-
Nearest gateway cluster	Section 5.25.2.1 ZbZclNearestGwClientAlloc	Instantiate a new instance of the Nearest Gateway client cluster

	Function	Definition
Nearest gateway cluster	ZbZclNearestGwServerAlloc	Instantiate a new instance of the Nearest Gateway server cluster
	ZbZclNearestGwServerAttrT	Nearest Gateway Attribute Ids
Occupancy sensing cluster	ZbZclOccupancyClientAlloc	Create a new instance of the Occupancy Sensing Server cluster
	ZbZclOccupancyServerAlloc	Create a new instance of the Occupancy Sensing Client cluster
	ZbZclOccupancySvrAttrT	-
On/Off switch configuration cluster	ZbZclOnOffSwConfigClientAlloc	Create a new instance of the OnOff Switch Configuration Client cluster
	ZbZclOnOffSwConfigServerAlloc	Create a new instance of the OnOff Switch Configuration Server cluster
	ZbZclOnOffSwConfigSvrAttrId	-
On/Off cluster	ZbZclOnOffClientAlloc	Create a new instance of the On/Off Client cluster
	ZbZclOnOffClientOffReq	Send an Off command
	ZbZclOnOffClientOnReq	Send an On command
	ZbZclOnOffClientToggleReq	Send a Toggle command
	ZbZclOnOffServerAlloc	Create a new instance of the OnOff Server cluster
	ZbZclOnOffSvrAttrT	-
	ZbZclOnOffServerCallbacksT	OnOff Server callbacks configuration
Over-The-Air upgrade cluster	ZbZclOtaClientAlloc	Create a new instance of the OTA Upgrade Client cluster
	ZbZclOtaClientDiscover	Discover OTA Upgrade Server
	ZbZclOtaClientDiscoverForced	Set the OTA Upgrade Server directly (without discovery)
	ZbZclOtaClientDiscoverForced	Set the OTA Upgrade Server directly (without discovery)
	ZbZclOtaClientGetDefaultCallbacks	Load the default callbacks for ECDSA Suite 2 support
	ZbZclOtaClientImageTransferResume	Resume an OTA Upgrade transfer `ZbZclOtaClientImageTransferResume` is only called by the NHLE if it returned ZCL_STATUS_WAIT_FOR_DATA for the write_`image` callback.
	ZbZclOtaClientImageTransferStart	Initiate a OTA Upgrade transfer
	ZbZclOtaClientQueryNextImageReq	Send a Query Next Image Request command
	ZbZclOtaHeaderParse	Parse an OTA Upgrade payload buffer’s header information
	ZbZclOtaServerAlloc	Create a new instance of the OTA Upgrade Server cluster
	ZbZclOtaServerImageNotifyReq	Send an OTA image notify server command registering an image. This does not automatically send an image notify message, the OTA server application can use `ZbZclOtaServerImageNotifyReq` after registering an image to notify clients of the availability of a new image
	ZbZclOtaServerUpgradeEndResp	Send an OTA upgrade end response

	Function	Definition
Over-The-Air upgrade cluster	ZbZclOtaActivationPolicy	-
	ZbZclOtaHeaderFieldCtrlBitmask	-
	ZbZclOtaImageBlkReqFldCtrl	-
	ZbZclOtaImageNotifyCmd	-
	ZbZclOtaImageType	-
	ZbZclOtaQueryFldCtrlHwVer	-
	ZbZclOtaSecCredential	-
	ZbZclOtaStackVersion	-
	ZbZclOtaStatus	-
	ZbZclOtaSubElementTag	-
	ZbZclOtaSvrAttrId	-
	ZbZclOtaTimeoutPolicy	-
	ZbZclOtaClientCallbacksT	OTA Upgrade callbacks configuration
	ZbZclOtaClientConfig	OTA Upgrade Client Configuration structure
	ZbZclOtaEndResponseTimes	Upgrade End Response command structure
	ZbZclOtaHeader	OTA Upgrade Header Fields structure
	ZbZclOtaImageData	OTA Upgrade Image Data structure
	ZbZclOtaImageDefinition	OTA Header Image Definition structure
	ZbZclOtaImageWaitForData	Image Block Response Command Payload with WAIT_FOR_DATA status structure
	ZbZclOtaServerConfig	OTA Upgrade Server Configuration structure
Poll control cluster	zcl_poll_client_alloc	Create a new instance of the Poll Control Client cluster
	zcl_poll_client_set_checkin_rsp	Set Check-in Response configuration
	zcl_poll_client_set_long_intvl_req	Send a Set Long Interval command
	zcl_poll_client_set_short_intvl_req	Send a Set Short Interval command
	zcl_poll_client_stop_fastpoll_req	Send a Fast Poll Stop command
	zcl_poll_server_alloc	Create a new instance of the Poll Control Server cluster
	zcl_poll_server_send_checkin	Send a Check-in command to any bound Clients.
	zcl_poll_server_write_long_poll_intvl	Change the long polling interval. Default is disabled (ZCL_INVALID_UNSIGNED_32BIT = 0xFFFFFFFF).
	ZbZclPollControlSvrAttrT	-
	ZbZclPollControlClientCallbackT	Poll Control Client callbacks configuration
	ZbZclPollControlClientCheckinInfo	Check-in Info structure
	ZbZclPollControlClientSetLongReq	Set Long Poll Interval command structure
	ZbZclPollControlClientSetShortReq	Set Short Poll Interval command structure
	ZbZclPollControlClientStopReq	Fast Poll Stop command structure
	ZbZclPollControlServerCallbackT	Poll Control Server callbacks configuration
	zcl_poll_checkin_rsp_t	Check-in Response command structure
Power configuration cluster	ZbZclPowerConfigClientAlloc	Create a new instance of the Power Configuration Client cluster

	Function	Definition
Power configuration cluster	ZbZclPowerConfigServerAlloc	Create a new instance of the Power Configuration Server cluster
	ZbZclPowerConfigSvrAttrT	-
Power profile cluster	ZbZclPowerProfClientAlloc	Create a new instance of the Power Profile Client cluster
	ZbZclPowerProfClientPhasesNotify	Send an Energy Phases Schedule Notification command
	ZbZclPowerProfClientPhasesResponse	Send a `EnergyPhasesScheduleResponse` command
	ZbZclPowerProfClientPhasesSchedStateReq	Send a `EnergyPhasesScheduleStateRequest` command
	ZbZclPowerProfClientPriceExtRsp	Send a `GetPowerProfilePriceExtendedResponse` command
	ZbZclPowerProfClientPriceRsp	Send a `GetPowerProfilePriceResponse` command
	ZbZclPowerProfClientProfileReq	Send a `PowerProfileRequest` command
	ZbZclPowerProfClientSchedConsReq	Send a `PowerProfileScheduleConstraintsRequest` command
	ZbZclPowerProfClientSchedPriceRsp	Send a `GetOverallSchedulePriceResponse` command
	ZbZclPowerProfClientProfileReq	Send a `PowerProfileRequest` command
	ZbZclPowerProfClientSchedConsReq	Send a `PowerProfileScheduleConstraintsRequest` command
	ZbZclPowerProfClientSchedPriceRsp	Send a `GetOverallSchedulePriceResponse` command
	ZbZclPowerProfServerGetPriceReq	Send a `GetPowerProfilePrice` command
	ZbZclPowerProfClientStateReq	Send a PowerProfileStateRequest command
	ZbZclPowerProfServerAlloc	Create a new instance of the Power Profile Server cluster
	ZbZclPowerProfServerConstraintsNotify	Send a `PowerProfileScheduleConstraintsNotification` command
	ZbZclPowerProfServerConstraintsRsp	Send a `PowerProfileScheduleConstraintsResponse` command
	ZbZclPowerProfServerGetPriceReq	Send a `GetPowerProfilePrice` command
	ZbZclPowerProfServerGetPriceReqExtReq]	Send a `GetPowerProfilePriceExtended` command
	ZbZclPowerProfServerPhasesNotify	Send a `EnergyPhasesScheduleStateNotification` Command

	Function	Definition
Power profile cluster	ZbZclPowerProfServerPhasesReq	Send a EnergyPhasesScheduleRequest command
	ZbZclPowerProfServerPhasesRsp	Send a EnergyPhasesScheduleStateResponse command
	ZbZclPowerProfServerProfileNotify	Send a PowerProfileNotification command. It is sent as a ZCL request with a unique sequence number, and can receive a default response if applicable
	ZbZclPowerProfServerProfileRsp	Send a PowerProfileResponse command
	ZbZclPowerProfServerProfileNotify	Send a PowerProfileNotification command. It is sent as a ZCL request with a unique sequence number, and can receive a default response if applicable
	ZbZclPowerProfServerProfileRsp	Send a PowerProfileResponse command
	ZbZclPowerProfServerStateNotify	Send a PowerProfileStateNotification Command
	ZbZclPowerProfServerStateRsp	Send a PowerProfileStateResponse command
	ZbZclPowerProfileSvrAttrT	-
	ZbZclPowerProfileSvrAttrT	Send a PowerProfileStateResponse command
	ZbZclPowerProfCliPhasesNotify	Energy Phases Schedule Notification command structure
	ZbZclPowerProfCliPriceRsp	GetPowerProfilePriceResponse command structure
	ZbZclPowerProfCliProfileReq	PowerProfileRequest command structure
	ZbZclPowerProfCliSchedPriceRsp	GetOverallSchedulePriceResponse command structure
	ZbZclPowerProfClientCallbacks	Power Profile Client callbacks configuration
	ZbZclPowerProfPhase	Phase structure
	ZbZclPowerProfSchedPhase	Schedule Phase structure
	ZbZclPowerProfServerCallbacks	Power Profile Server callbacks configuration
	ZbZclPowerProfSvrConstraintsNotify	PowerProfileScheduleConstraintsNotification command structure
	ZbZclPowerProfSvrGetPriceExtReq	GetPowerProfilePriceExtended command structure
	ZbZclPowerProfSvrPhasesRsp	Energy Phases Schedule Notification and EnergyPhasesScheduleResponse command structure
	ZbZclPowerProfSvrProfileRsp	PowerProfileNotification and PowerProfileResponse command structure
	ZbZclPowerProfSvrStateRsp	PowerProfileStateResponse command structure
	ZbZclPowerProfileRecord	Record structure
Pressure measurement cluster	ZbZclPressMeasClientAlloc	Create a new instance of the Pressure Measurement Client cluster

	Function	Definition
Pressure measurement cluster	ZbZclPressMeasServerAlloc	Create a new instance of the Pressure Measurement Server cluster
	ZbZclPressMeasSvrAttrT	-
Price cluster	ZbZclPriceClientAlloc	Create a new instance of the Price Client cluster
	ZbZclPriceClientCommandGetBlockThresholdsReq	Send a Get Block Thresholds command
	ZbZclPriceClientCommandGetCurrentPriceReq	Send a Get Current Price command
	ZbZclPriceClientCommandGetPriceMatrixReq	Send a Get Price Matrix command
	ZbZclPriceClientCommandGetScheduledPricesReq	Send a Get Scheduled Prices command
	ZbZclPriceClientCommandGetTariffInfoReq	Send a Get Tariff Information command
	ZbZclPriceClientCommandPriceAckReq	Send a Price Acknowledgement command
	ZbZclPriceServerAlloc	Create a new instance of the Price Server cluster
	ZbZclPriceServerPublishPriceInit	Initialize Publish Price information
	ZbZclPriceServerSendPublishBlockThresholds	Send a Publish Block Thresholds command
	ZbZclPriceServerSendPublishMatrix	Send a Publish Price Matrix command
	ZbZclPriceServerSendPublishPrice	Send a Publish Price command
	ZbZclPriceServerSendPublishTariffInfo	Send a Publish Tariff Information command
	ZbZclPriceSvrAttrT	-
	ZbZclPriceClientCallbacksT	Price Client callbacks configuration
	ZbZclPriceClientGetBlockThresholdsT	Get Block Thresholds command structure
	ZbZclPriceClientGetCurrentPriceT	Get Current Price command structure
	ZbZclPriceClientGetPriceMatrixT	Get Price Matrix command structure
	ZbZclPriceClientGetScheduledPricesT	Get Scheduled Prices command structure
	ZbZclPriceClientGetTariffInfoT	Get Tariff Information command structure
	ZbZclPriceClientPriceAckT	Price Acknowledge command structure
	ZbZclPriceServerBlockThreshEntryT	Block Threshold Entry structure
	ZbZclPriceServerCallbacksT	Price Server callbacks configuration
	ZbZclPriceServerPriceMatrixEntryT	Price Matrix Entry structure
	ZbZclPriceServerPublishBlockThresholdsT	Publish Block Thresholds command structure
	ZbZclPriceServerPublishPriceMatrixT	Publish Price Matrix command structure
	ZbZclPriceServerPublishPriceT	Publish Price command structure
	ZbZclPriceServerPublishTariffInfoT	Publish Tariff Info command structure
Pump configuration and control cluster	ZbZclPumpClientAlloc	Instantiate a new instance of the Pump Configuration and Control client cluster
	ZbZclPumpServerAlloc	Instantiate a new instance of the Pump Configuration and Control server cluster
	ZbZclPumpServerAttrT	-
RSSI location cluster	ZbZclRssiLocClientAlloc	Create a new instance of the RSSI Location Client cluster
	ZbZclRssiLocClientAnchorNodeAnnc	Send a Anchor Node Announce command
	ZbZclRssiLocClientGetDevConfig	Send a Get Device Configuration command
	ZbZclRssiLocClientGetLocData	Send a Get Location Data command
	ZbZclRssiLocClientSendPings	Send a Send Pings command

	Function	Definition
RSSI location cluster	ZbZclRssiLocClientSendRssiRsp	Send a RSSI Response command
	ZbZclRssiLocClientSetAbsLocation	Send a Set Absolute Location command
	ZbZclRssiLocClientSetDevConfig	Send a Set Device Configuration command
	ZbZclRssiLocServerAlloc	Create a new instance of the RSSI Location Server cluster
	ZbZclRssiLocServerCompDataNotif	Send a Compact Data Notification command
	ZbZclRssiLocServerLocDataNotif	Send a Location Data Notification command
	ZbZclRssiLocServerReportRssi	Send a Report RSSI command
	ZbZclRssiLocServerReqOwnLoc	Send a Request Own Location command
	ZbZclRssiLocServerRssiPing	Send a RSSI Ping command
	ZbZclRssiLocServerRssiReq	Send a RSSI Request command
	ZbZclRssiLocServerSendDevConfigRsp	Send a Device Configuration Response command
	ZbZclRssiLocServerSendLocDataRsp	Send a Location Data Response command
	ZbZclRssiLocSvrAttrT	-
	rssi_loc_anchor_node_annc	Anchor Node Announce command structure
	rssi_loc_comp_data_notif	Compact Data Notification command structure
	rssi_loc_dev_config_rsp	Device Configuration response structure
	rssi_loc_get_dev_config	Get Device Configuration command structure
	rssi_loc_get_loc_data	Get Location Data command structure
	rssi_loc_loc_data_notif	Location Data Notification command structure
	rssi_loc_loc_data_rsp	Location Data response structure
	rssi_loc_neighbour_info	Neighbor Info structure
	rssi_loc_report_rssi	Report RSSI command structure
	rssi_loc_req_own_loc	Request Own Location command structure
	rssi_loc_rssi_ping	RSSI Ping command structure
	rssi_loc_rssi_req	RSSI Request structure
	rssi_loc_rssi_rsp	RSSI response structure
	rssi_loc_send_pings	Send Pings command structure
	rssi_loc_set_abs_loc	Set Absolute Location command structure
	rssi_loc_set_dev_config	Set Device Configuration command structure
	zcl_rssi_loc_client_callbacks_t	RSSI Location Client callbacks configuration
	zcl_rssi_loc_server_callbacks_t	RSSI Location Server callbacks configuration
Scenes cluster	ZbZclScenesClientAlloc	Create a new instance of the Scenes Client cluster
	ZbZclScenesServerAlloc	Create a new instance of the Scenes Server cluster
	zcl_scenes_client_add_req	Send an Add Scene or Enhanced Add Scene command, depending on is Enhanced flag in Add Scene command structure
	zcl_scenes_client_add_rsp_parse	Parse an Add Scene Response command payload into a structure
	zcl_scenes_client_copy_req	Send a Copy Scene command

	Function	Definition
Scenes cluster	zcl_scenes_client_copy_rsp_parse	Parse a Copy Scene Response command payload into a structure
	zcl_scenes_client_get_membership_req	Send a Get Scene Membership command
	zcl_scenes_client_get_membership_rsp_parse	Parse a Get Scene Membership Response command payload into a structure
	zcl_scenes_client_recall_req	Send a Recall Scene command
	zcl_scenes_client_recall_rsp_parse	Parse a Recall Scene Response command payload into a structure
	zcl_scenes_client_remove_all_req	Send a Remove All Scenes command
	zcl_scenes_client_remove_all_rsp_parse	Parse a Remove All Scenes Response command payload into a structure
	zcl_scenes_client_remove_req	Send a Remove Scene command
	zcl_scenes_client_remove_rsp_parse	Parse a Remove Scene Response command payload into a structure
	zcl_scenes_client_store_req	Send a Store Scene command
	zcl_scenes_client_store_rsp_parse	Parse a Store Scene Response command payload into a structure
	zcl_scenes_client_view_req	Send an View Scene or Enhanced View Scene command, depending on is Enhanced flag in View Scene command structure
	zcl_scenes_client_view_rsp_parse	Parse a View Scene Response command payload into a structure
	ZbZclScenesAttrT	-
	zcl_scenes_add_request_t	Add Scene command structure
	zcl_scenes_add_response_t	Add Scene Response command structure
	zcl_scenes_copy_request_t	Copy Scene command structure
	zcl_scenes_copy_response_t	Copy Scene Response command structure
	zcl_scenes_membership_request_t	Get Scene Membership command structure
	zcl_scenes_membership_response_t	Get Scene Membership Response command structure
	zcl_scenes_recall_request_t	Recall Scene command structure
	zcl_scenes_recall_response_t	Recall Scene Response command structure
	zcl_scenes_remove_all_request_t	Remove All Scenes command structure
	zcl_scenes_remove_all_response_t	Remove All Scenes Response command structure
	zcl_scenes_remove_request_t	Remove Scene command structure
	zcl_scenes_store_response_t	Store Scene Response command structure
	zcl_scenes_view_request_t	View Scene command structure
	zcl_scenes_view_response_t	View Scene Response command structure
Temperature measurement cluster	ZbZclTempMeasClientAlloc	Create a new instance of the Temperature Measurement Client cluster
	ZbZclTempMeasServerAlloc	Create a new instance of the Temperature Measurement Server cluster
	ZbZclTempMeasSvrAttrT	-
Thermostat cluster	ZbZclThermClientAlloc	Create a new instance of the Scenes Client cluster

	Function	Definition
Thermostat cluster	ZbZclThermClientClearWeeklySched	Send a Clear Weekly Schedule command (optional)
	ZbZclThermClientGetRelayStatusLog	Send a Get Relay Status Log command (optional)
	ZbZclThermClientGetWeeklySched	Send a Get Weekly Schedule command (optional)
	ZbZclThermClientSetWeeklySched	Send a Set Weekly Schedule command (optional)
	ZbZclThermClientSetpointRaiseLower	Send a Setpoint Raise/Lower command
	ZbZclThermServerAlloc	Create a new instance of the Thermostat Server cluster
	ZbZclThermServerGetRelayStatusLogRsp	Send a Get Relay Status Log Response command
	ZbZclThermServerGetWeeklySchedRsp	Send a Get Weekly Schedule Response command
	ZbZclThermAttrT	-
	ZbZclThermCliGetWeeklyT	Thermostat Get Weekly Schedule structure
	ZbZclThermCliSetpointT	Thermostat Setpoint structure
	ZbZclThermServerCallbacksT	Thermostat Server callbacks configuration
	ZbZclThermSvrGetRelayStatusLogRspT	Get Relay Status Log Response structure
	ZbZclThermTransitionsT	Thermostat Transition structure
	ZbZclThermWeeklySchedT	Thermostat Weekly Schedule structure
Thermostat user interface cluster	ZbZclThermUiClientAlloc	Create a new instance of the Thermostat User Interface Client cluster
	ZbZclThermUiServerAlloc	Create a new instance of the Thermostat User Interface Server cluster
	ZbZclThermUiServerAttrT	-
Time cluster	ZbZclTimeClientAlloc	Create a new instance of the Time Client cluster
	ZbZclTimeServerAlloc	Create a new instance of the Time Server cluster
	ZbZclTimeServerCurrentTime	Call the get_time callback defined as part of the Time Server callbacks configuration
	ZbZclTimeServerSetTime	Call the set_time callback defined as part of the Time Server callbacks configuration
	ZbZclTimeSvrAttrT	-
	ZbZclTimeServerCallbacks	Time Server callbacks configuration
Tunneling cluster	ZbZclTunnelClientAddProto	Add a protocol tunnel to the server
	ZbZclTunnelClientAlloc	Create a new instance of the Tunneling Client cluster
	ZbZclTunnelClientCloseQuietReq	Close the local tunnel without informing the server
	ZbZclTunnelClientCloseReq	Send a Close Tunnel command
	ZbZclTunnelClientConnectReq	Send a Request Tunnel command
	ZbZclTunnelClientSendReq	Send a Transfer Data command
	ZbZclTunnelServerAddProto	Add a protocol tunnel to the server
	ZbZclTunnelServerAlloc	Create a new instance of the Tunneling Server cluster
	ZbZclTunnelServerSendAllMatch	Send data to all tunnel clients with matching EUI

	Function	Definition
Tunneling cluster	ZbZclTunnelServerSendto	Send data via the tunnel
	ZbZclTunnelServerStateFindById	Get tunnel server state by tunnel ID
	ZbZclTunnelStateGetDataLen	Get received data length using Tunneling Cluster State structure
	ZbZclTunnelStateGetDataPtr	Get received data using Tunneling Cluster State structure
	ZbZclTunnelStateGetId	Get ID using Tunneling Cluster State structure
	ZbZclTunnelStateGetProtocol	Get Tunneling Protocol Instance using Tunneling Cluster State structure
	ZbZclTunnelProtocolT	-
	ZbZclTunnelStatusT	-
	ZbZclTunnelSvrAttrT	-
	ZbZclTunnelXferStatusT	-
Touchlink cluster	ZbZclTouchlinkInitiatorGetEpListReq	Sends a Touchlink Utility Get Endpoint List Request Command
	ZbZclTouchlinkInitiatorGetGrpIdReq	Sends a Touchlink Utility Get Group Identifiers Request Command
	ZbZclTouchlinkTargetSendEpInfoCmd	Sends a Touchlink Utility Endpoint Information Command
	ZbTlEndpointList	Endpoint Information Record Entry
	ZbTlEpInfoCmd	Endpoint Information Command
	ZbTlGetEpListReqCmd	Get Endpoint List Request command
	ZbTlGetEpListRspCmd	Get Endpoint List Response Command
	ZbTlGetGroupIdsReqCmd	Get Group Identifiers Request command
	ZbTlGetGroupIdsRspCmd	Get Group Identifiers Response command
	ZbTlGroupRecordList	Group Information Record list
	ZbTouchlinkCallbacks	Zigbee Touchlink callback functions for Touchlink Controller Device Utility commands. These are configured by "struct ZbStartupT" with `ZbStartup()`.
Voice over Zigbee® cluster	ZbZclVoiceClientAlloc	Create a new instance of the Voice Over Zigbee Client cluster
	ZbZclVoiceClientEstabReq	Send an Establishment Request command
	ZbZclVoiceClientSendControlRsp	Send an Control command
	ZbZclVoiceServerAlloc	Create a new instance of the Voice Over Zigbee Server cluster
	ZbZclVoiceServerControlReq	Send a Control command
	ZbZclVoiceServerSendEstabRsp	Send an Establishment Response command
	ZbZclVoiceServerSendVoiceTxRsp	Send a Voice Transmission Response command. The application calls this if it ever encounters an error processing a Voice Transmission packet
	ZbZclVoiceTxCompletedReq	Send a Voice Transmission Complete command
	ZbZclVoiceVoiceTxReq	Send a Voice Transmission command
	ZbZclVoiceSvrAttrT	-
Water content measurement cluster	ZbZclWaterContentMeasClientAlloc	Create a new instance of the Water Content Measurement Client cluster

	Function	Definition
Water content measurement cluster	ZbZclWaterContentMeasServerAlloc	Create a new instance of the Water Content Measurement Server cluster
	ZbZclWcmSvrAttrT	-
Window covering cluster	ZbZclWindowClientAlloc	Create a new instance of the Window Covering Client cluster
	ZbZclWindowClientCommandDown	Send a Down/Close command
	ZbZclWindowClientCommandStop	Send a Stop command
	ZbZclWindowClientCommandUp	Send an Up/Open command
	ZbZclWindowClosureServerMode	Configure the Window Covering mode
	ZbZclWindowServerAlloc	Create a new instance of the Window Covering Server cluster
	ZbZclWncvServerAttrT	-
	Section 5.46.3.2 ZbZclWncvTypes	-
	ZbZclWindowServerCallbacksT	Window Covering Server callbacks configuration

5.2 Alarms cluster

```
#include "zcl/general/zcl.alarm.h"
```

5.2.1 Alarms cluster description

Table 8. **Alarms cluster PICS code description**

PICS Code	Name	Supported	Notes
ALM.S	Server	True	-
ALM.C	Client	True	-
ALM.S.A0000	`AlarmCount` server attribute	True	Optional
ALM.S.Afffd	`ClusterRevision` server attribute	True	-
ALM.S.Afffe	`AttributeReportingStatus` server attribute	False	-
ALM.S.C00.Rsp	Reset Alarm server command	True	-
ALM.S.C01.Rsp	Reset all alarms server command	True	-
ALM.S.C02.Rsp	Get Alarm server command	True	-
ALM.S.C03.Rsp	Reset alarm log server command	True	-
ALM.S.C00.Tx	Alarm server command	True	-
ALM.S.C01.Tx	Get alarm response server command	True	-
ALM.C.Afffd	`ClusterRevision` client attribute	True	-
ALM.C.Afffe	`AttributeReportingStatus` client attribute	False	-
ALM.C.C00.Rsp	Alarm client command	True	-
ALM.C.C01.Rsp	Get alarm response client command	True	-
ALM.C.C00.Tx	Reset Alarm client command	True	-
ALM.C.C01.Tx	Reset all alarms client command	True	-
ALM.C.C02.Tx	Get Alarm client command	True	-
ALM.C.C03.Tx	Reset alarm log client command	True	-

5.2.2 Alarms cluster functions

5.2.2.1 *ZbZclAlarmClientAlloc*

```
struct ZbZclClusterT * ZbZclAlarmClientAlloc(struct ZigBeeT *zb, uint8_t endpoint,
ZbZclAlarmClientCallbackT callback, void *arg);
```

Create a new instance of the Alarms Client cluster.

Table 9. **ZbZclAlarmClientAlloc parameters**

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: Cluster pointer, or NULL if there is an error.

5.2.2.2 *ZbZclAlarmClientGetAlarmReq*

```
enum ZclStatusCodeT ZbZclAlarmClientGetAlarmReq(struct ZbZclClusterT *cluster, const struct
ZbApsAddrT *dst, void (*callback)(struct ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Get Alarm command.

Table 10. **ZbZclAlarmClientGetAlarmReq parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.2.2.3 *ZbZclAlarmClientResetAlarmLogReq*

```
enum ZclStatusCodeT ZbZclAlarmClientResetAlarmLogReq(struct ZbZclClusterT *cluster, const
struct ZbApsAddrT *dst, void (*callback)(struct ZbZclCommandRspT *rsp, void *arg), void
*arg);
```

Send a Reset Alarm Log command.

Table 11. **ZbZclAlarmClientResetAlarmLogReq parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.2.2.4 ZbZclAlarmClientResetAlarmReq

```
enum ZclStatusCodeT ZbZclAlarmClientResetAlarmReq(struct ZbZclClusterT *cluster, const
struct ZbApsAddrT *dst, uint8_t alarm_code, uint16_t cluster_id, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Reset Alarm command

Table 12. ZbZclAlarmClientResetAlarmReq parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`alarm_code`	Code of the detected alarm condition
`cluster_id`	ID of cluster where alarm condition occurred
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.2.2.5 ZbZclAlarmClientResetAllAlarmsReq

```
enum ZclStatusCodeT ZbZclAlarmClientResetAllAlarmsReq(struct ZbZclClusterT *cluster, const
struct ZbApsAddrT *dst, void (*callback)(struct ZbZclCommandRspT *rsp, void *arg), void
*arg);
```

Send a Reset All Alarms command

Table 13. ZbZclAlarmClientResetAllAlarmsReq parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.2.2.6 ZbZclAlarmServerAlloc

```
struct ZbZclClusterT * ZbZclAlarmServerAlloc(struct ZigBeeT *zb, uint8_t endpoint, uint16_t
logSize, struct ZbZclClusterT *time_server);
```

Create a new instance of the Alarms Server cluster.

Table 14. ZbZclAlarmServerAlloc parameters

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster
`logSize`	Alarm log size
`time_server`	Time server cluster instance used to retrieve event timing information

Return: Cluster pointer, or NULL if there is an error.

5.2.3 Alarms cluster enumerations

5.2.3.1 *ZbZclAlarmsAttrT*

Alarms attribute IDs

Table 15. ZbZclAlarmsAttrT attribute

Alarm attribute	Description
ZCL_ALARM_ATTR_COUNT	`AlarmCount` (optional)

5.3 Ballast configuration cluster

```
#include "zcl/general/zcl.ballast.config.h"
```

5.3.1 Ballast configuration cluster description

Table 16. **Ballast configuration cluster PICS code description**

PICS Code	Name	Supported	Notes
BC.S	Server	True	-
BC.C	Client	True	-
BC.S.A0000	`PhysicalMinLevel` server attribute	True	-
BC.S.A0001	`PhysicalMaxLevel` server attribute	True	-
BC.S.A0002	`BallastStatus` server attribute	True	Optional
BC.S.A0010	`MinLevel` server attribute	True	-
BC.S.A0011	`MaxLevel` server attribute	True	-
BC.S.Afffd	`ClusterRevision` server attribute	True	-
BC.S.Afffe	`AttributeReportingStatus` server attribute	False	-
BC.C.Afffd	`ClusterRevision` client attribute	True	-
BC.C.Afffe	`AttributeReportingStatus` client attribute	False	-

5.3.2 Ballast configuration cluster functions

5.3.2.1 *ZbZclBallastConfigClientAlloc*

```
struct ZbZclClusterT * ZbZclBallastConfigClientAlloc(struct ZigBeeT *zb, uint8_t endpoint);
```

Create a new instance of the ballast configuration client cluster

Table 17. **ZbZclBallastConfigClientAlloc Parameters**

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster

Return: Cluster pointer, or NULL if there is an error.

5.3.2.2 *ZbZclBallastConfigServerAlloc*

```
struct ZbZclClusterT * ZbZclBallastConfigServerAlloc(struct ZigBeeT *zb, uint8_t endpoint,
uint8_t phyMin, uint8_t phyMax);
```

Create a new instance of the Ballast Configuration Server cluster

Table 18. **ZbZclBallastConfigServerAlloc parameters**

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster
`phyMin`	The default minimum light output
`phyMax`	The default maximum light output

Return: Cluster pointer, or NULL if there is an error.

5.3.3 Ballast configuration cluster enumerations

5.3.3.1 *ZbZclBallastConfigSvrAttrT*

Table 19. **Ballast Configuration server attributes IDs**

Attribute ID	Description
ZCL_BALLAST_CONFIG_ATTR_PHY_MIN_LEVEL	`PhysicalMinLevel`
ZCL_BALLAST_CONFIG_ATTR_PHY_MAX_LEVEL	`PhysicalMaxLevel`
ZCL_BALLAST_CONFIG_ATTR_BALLAST_STATUS	`BallastStatus`
ZCL_BALLAST_CONFIG_ATTR_MIN_LEVEL	`MinLevel`
ZCL_BALLAST_CONFIG_ATTR_MAX_LEVEL	`MaxLevel`
ZCL_BALLAST_CONFIG_ATTR_POWER_ON_LEVEL	`PowerOnLevel` (deprecated)
ZCL_BALLAST_CONFIG_ATTR_POWER_ON_FADE_T IME	`PowerOnFadeTime` (deprecated)
ZCL_BALLAST_CONFIG_ATTR_INTRINSIC_BALLAST7_FACTOR	`IntrinsicBallastFactor` (optional)
ZCL_BALLAST_CONFIG_ATTR_BALLAST_FACTOR_A DJUSTMENT	`BallastFactorAdjustment` (optional)
ZCL_BALLAST_CONFIG_ATTR_LAMP_QUANTITY	`LampQuantity` (optional)
ZCL_BALLAST_CONFIG_ATTR_LAMP_TYPE	`LampType` (optional)
ZCL_BALLAST_CONFIG_ATTR_LAMP_MANUFACTUR ER	`LampManufacturer` (optional)
ZCL_BALLAST_CONFIG_ATTR_LAMP_RATED_HOUR S	`LampRatedHours` (optional)
ZCL_BALLAST_CONFIG_ATTR_LAMP_BURN_HOURS	`LampBurnHours` (optional)
ZCL_BALLAST_CONFIG_ATTR_LAMP_ALARM_MODE	`LampAlarmMode` (optional)
ZCL_BALLAST_CONFIG_ATTR_LAMP_BURN_HOURS_TRIP_POINT	`LampBurnHoursTripPoint` (optional)

5.4 Basic cluster

```
#include "zcl/general/zcl.basic.h"
```

5.4.1 Basic cluster description

Table 20. **Basic cluster PICS code description**

PICS Code	Name	Supported	Notes
B.S	Server	True	-
B.C	Client	True	-
B.S.A0000	`ZCLVersion` Server attribute	True	-
B.S.A0001	`ApplicationVersion` Server attribute	True	Optional
B.S.A0002	`StackVersion` Server attribute	True	Optional
B.S.A0003	`HWVersion` Server attribute	True	Optional
B.S.A0004	`ManufacturerName` Server attribute	True	Optional
B.S.A0005	`ModelIdentifier` Server attribute	True	Optional
B.S.A0006	`DateCode` Server attribute	True	Optional
B.S.A0007	`PowerSource` Server attribute	True	-

PICS Code	Name	Supported	Notes
B.S.A0010	`LocationDescription` Server attribute	True	Optional
B.S.A0011	`PhysicalEnvironment` Server attribute	True	Optional
B.S.A0013	`AlarmMask` Server attribute	True	Optional
B.S.A0014	`DisableLocalConfig` Server attribute	True	Optional
B.S.A4000	`SWBuildID` Server attribute	True	Optional
B.S.Afffd	`ClusterRevision` Server attribute	True	-
B.S.Afffe	`AttributeReportingStatus` Server attribute	False	-
B.S.C00.Rsp	Reset to Factory Defaults server command	True	-
B.C.Afffd	`ClusterRevision` client attribute	True	-
B.C.Afffe	`AttributeReportingStatus` client attribute	False	-
B.C.C00.Tx	Reset to Factory Defaults client command	True	-

5.4.2 Basic cluster functions

5.4.2.1 *ZbZclBasicClientAlloc*

```
struct ZbZclClusterT * ZbZclBasicClientAlloc(struct ZigBeeT *zb, uint8_t endpoint);
```

Create a new instance of the Basic Client cluster

Table 21. **ZbZclBasicClientAlloc Parameters**

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster

Return: Cluster pointer, or NULL if there is an error.

5.4.2.2 *ZbZclBasicClientResetReq*

```
enum ZclStatusCodeT ZbZclBasicClientResetReq(struct ZbZclClusterT *cluster, const struct
ZbApsAddrT *dst);
```

Send a Reset to Factory Defaults command

Table 22. **ZbZclBasicClientResetReq parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.4.3 Basic cluster enumerations

5.4.3.1 *ZbZclBasicSvrAttrT*

Table 23. ZbZclBasicSvrAttrT basic Server attribute IDs

Attribute ID	Description
ZCL_BASIC_ATTR_ZCL_VERSION	ZCLVersion
ZCL_BASIC_ATTR_APP_VERSION	ApplicationVersion (optional)
ZCL_BASIC_ATTR_STACK_VERSION	StackVersion (optional)
ZCL_BASIC_ATTR_HARDWARE_VERSION	HWVersion (optional)
ZCL_BASIC_ATTR_MFR_NAME	ManufacturerName (optional)
ZCL_BASIC_ATTR_MODEL_NAME	ModelIdentifier (optional)
ZCL_BASIC_ATTR_DATE_CODE	DateCode (optional)
ZCL_BASIC_ATTR_POWER_SOURCE	PowerSource
ZCL_BASIC_ATTR_LOCATION	LocationDescription (optional)
ZCL_BASIC_ATTR_ENVIRONMENT	PhysicalEnvironment (optional)
ZCL_BASIC_ATTR_ENABLED	DeviceEnabled (optional)
ZCL_BASIC_ATTR_ALARM_MASK	AlarmMask (optional)
ZCL_BASIC_ATTR_DISABLE_LOCAL_CONFIG	DisableLocalConfig (optional)
ZCL_BASIC_ATTR_SW_BUILD_ID	SWBuildID (optional)

5.5 Color control cluster

```
#include "zcl/general/zcl.color.h"
```

5.5.1 Color control cluster description

Table 24. Color control cluster PICS code description

PICS Code	Name	Supported	Notes
CC.S	Server	True	-
CC.C	Client	True	-
CC.S.A0000	CurrentHue Server attribute	True	-
CC.S.A0000.Report.Tx	CurrentHue Server attribute Reports	True	-
CC.S.A0001	CurrentSaturation Server attribute	True	-
CC.S.A0001.Scene	CurrentSaturation Server attribute Scenes	True	-
CC.S.A0001.Report.Tx	CurrentSaturation Server attribute Reports	True	-
CC.S.A0003	CurrentX Server attribute	True	-
CC.S.A0003.Scene	CurrentX Server attribute scenes	True	-
CC.S.A0003.Report.Tx	CurrentX server attribute reports	True	-
CC.S.A0004	CurrentY server attribute	True	-
CC.S.A0004.Scene	CurrentY server attribute Scenes	True	-
CC.S.A0004.Report.Tx	CurrentY server attribute Reports	True	-
CC.S.A0007	ColorTemperatureMireds server attribute	True	-

PICS Code	Name	Supported	Notes
CC.S.A0007.Scene	ColorTemperatureMireds server attribute Scenes	True	-
CC.S.A0007.Report.Tx	ColorTemperatureMireds server attribute Reports	True	-
CC.S.A0008	ColorMode server attribute	True	-
CC.S.A000f	Options server attribute	True	-
CC.S.A0010	NumberOfPrimaries server attribute	True	-
CC.S.A0011	Primary1X server attribute	False	Mandatory if the value of the NumberOfPrimaries
CC.S.A0012	Primary1X server attribute	False	Mandatory if the value of the NumberOfPrimaries
CC.S.A0013	Primary1Intensity server attribute	False	Mandatory if the value of the NumberOfPrimaries
CC.S.A0015	Primary2X server attribute	False	Mandatory if the value of the NumberOfPrimaries
CC.S.A0016	Primary2Y server attribute	False	Mandatory if the value of the NumberOfPrimaries
CC.S.A0017	Primary2Intensity server attribute	False	Mandatory if the value of the NumberOfPrimaries
CC.S.A0019	Primary3X server attribute	False	Mandatory if the value of the NumberOfPrimaries
CC.S.A001a	Primary3Y server attribute	False	Mandatory if the value of the NumberOfPrimaries
CC.S.A001b	Primary3Intensity server attribute	False	Mandatory if the value of the NumberOfPrimaries
CC.S.A0020	Primary4X server attribute	False	Mandatory if the value of the NumberOfPrimaries
CC.S.A0021	Primary4Y server attribute	False	Mandatory if the value of the NumberOfPrimaries
CC.S.A0022	Primary4Intensity server attribute	False	Mandatory if the value of the NumberOfPrimaries
CC.S.A0024	Primary5X server attribute	False	Mandatory if the value of the NumberOfPrimaries
CC.S.A0025	Primary5Y server attribute	False	Mandatory if the value of the NumberOfPrimaries
CC.S.A0026	Primary5Intensity server attribute	False	Mandatory if the value of the NumberOfPrimaries
CC.S.A0028	Primary6X server attribute	False	Mandatory if the value of the NumberOfPrimaries

PICS Code	Name	Supported	Notes
CC.S.A0029	Primary6Y server attribute	False	Mandatory if the value of the NumberOfPrimaries
CC.S.A002a	Primary6Intensity server attribute	False	Mandatory if the value of the NumberOfPrimaries
CC.S.A4000	EnhancedCurrentHue server attribute	True	-
CC.S.A4000.Scene	EnhancedCurrentHue server attribute Scenes	True	-
CC.S.A4001	EnhancedColorMode server attribute	True	-
CC.S.A4002	ColorLoopActive server attribute	True	-
CC.S.A4002.Scene	ColorLoopActive server attribute Scenes	True	-
CC.S.A4003	ColorLoopDirection server attribute	True	-
CC.S.A4003.Scene	ColorLoopDirection server attribute Scenes	True	-
CC.S.A4004	ColorLoopTime server attribute	True	-
CC.S.A4004.Scene	ColorLoopTime server attribute Scenes	True	-
CC.S.A4005	ColorLoopStartEnhancedHue server attribute	True	-
CC.S.A4006	ColorLoopStartEnhancedHueColorLoopStoredEnhancedHue server attribute	True	-
CC.S.A400a	ColorCapabilities server attribute	True	-
CC.S.A400b	ColorTempPhysicalMinMireds server attribute	True	-
CC.S.A400c	ColorTempPhysicalMaxMireds server attribute	True	-
CC.S.A400d	ColorTempPhysicalMaxMireds server attribute	True	-
CC.S.A4010	ColorTemperatureMireds server attribute	True	-
CC.S.Afffd	`ClusterRevision` server attribute	True	-
CC.S.Afffe	`AttributeReportingStatus` server attribute	False	-
CC.S.C00.Rsp	Move to Hue server command	True	-
CC.S.C01.Rsp	Move Hue server command	True	-
CC.S.C02.Rsp	Step Hue server command	True	-
CC.S.C03.Rsp	Move to Saturation server command	True	-
CC.S.C04.Rsp	Move Saturation server command	True	-
CC.S.C05.Rsp	Step Saturation server command	True	-
CC.S.C06.Rsp	Move to Hue and Saturation server command	True	-
CC.S.C07.Rsp	Move to Color server command	True	-
CC.S.C08.Rsp	Move Color server command	True	-
CC.S.C09.Rsp	Step Color server command	True	-
CC.S.C0a.Rsp	Move to Color Temperature server command	True	-
CC.S.C40.Rsp	Enhanced Move to Hue server command	True	-
CC.S.C41.Rsp	Enhanced Move Hue server command	True	-
CC.S.C42.Rsp	Enhanced Step Hue server command	True	-
CC.S.C43.Rsp	Enhanced Move to Hue and Saturation server command	True	-
CC.S.C44.Rsp	Color Loop Set server command	True	-
CC.S.C47.Rsp	Stop Move Step server command	True	-
CC.S.C4b.Rsp	Move Color Temperature server command	True	-

PICS Code	Name	Supported	Notes
CC.S.C4c.Rsp	Step Color Temperature server command	True	-
CC.C.A0000.Report.Rsp	`CurrentHue` client attribute Reports	False	-
CC.C.A0001.Report.Rsp	`CurrentSaturation` client attribute Reports	False	-
CC.C.A0003.Report.Rsp	`CurrentX` client attribute Reports	False	-
CC.C.A0004.Report.Rsp	`CurrentY` client attribute reports	False	-
CC.C.A0007.Report.Rsp	`ColorTemperatureMireds` client attribute reports	False	-
CC.C.Afffd	`ClusterRevision` client attribute	True	-
CC.C.Afffe	`AttributeReportingStatus` client attribute	False	-
CC.C.C00.Tx	Move to Hue client command	True	-
CC.C.C01.Tx	Move Hue client command	True	-
CC.C.C02.Tx	Step Hue client command	True	-
CC.C.C03.Tx	Move to Saturation client command	True	-
CC.C.C04.Tx	Move Saturation client command	True	-
CC.C.C05.Tx	Step Saturation client command	True	-
CC.C.C06.Tx	Move to Hue and Saturation client command	True	-
CC.C.C07.Tx	Move to Color client command	True	-
CC.C.C08.Tx	Move Color client command	True	-
CC.C.C09.Tx	Step Color client command	True	-
CC.C.C0a.Tx	Move to Color Temperature client command	True	-
CC.C.C40.Tx	Enhanced Move to Hue client command	True	-
CC.C.C41.Tx	Enhanced Move Hue client command	True	-
CC.C.C42.Tx	Enhanced Step Hue client command	True	-
CC.C.C43.Tx	Enhanced Move to Hue and Saturation client command	True	-
CC.C.C44.Tx	Color Loop Set client command	True	-
CC.C.C47.Tx	Stop Move Step client command	True	-
CC.C.C4b.Tx	Move Color Temperature client command	True	-
CC.C.C4c.Tx	Step Color Temperature client command	True	-
CC.S.A0017	`Primary2Intensity` server attribute	False	Mandatory if the value of the NumberOfPrimaries
CC.S.A0019	`Primary3X` server attribute	False	Mandatory if the value of the NumberOfPrimaries
CC.S.A001a	`Primary3Y` server attribute	False	Mandatory if the value of the NumberOfPrimaries
CC.S.A001b	`Primary3Intensity` server attribute	False	Mandatory if the value of the NumberOfPrimaries
CC.S.A0020	`Primary4X` server attribute	False	Mandatory if the value of the NumberOfPrimaries
CC.S.A0021	`Primary4Y` server attribute	False	Mandatory if the value of the NumberOfPrimaries

PICS Code	Name	Supported	Notes
CC.S.A0022	`Primary4Intensity` server attribute	False	Mandatory if the value of the NumberOfPrimaries
CC.S.A0024	`Primary5X` server attribute	False	Mandatory if the value of the NumberOfPrimaries
CC.S.A0025	`Primary5Y` server attribute	False	Mandatory if the value of the NumberOfPrimaries
CC.S.A0026	`Primary5Intensity` server attribute	False	Mandatory if the value of the NumberOfPrimaries
CC.S.A0028	`Primary6X` server attribute	False	Mandatory if the value of the NumberOfPrimaries
CC.S.A0029	`Primary6Y` server attribute	False	Mandatory if the value of the NumberOfPrimaries
CC.S.A002a	`Primary6Intensity` server attribute	False	Mandatory if the value of the NumberOfPrimaries
CC.S.A4000	`EnhancedCurrentHue` server attribute	True	-
CC.S.A4000.Scene	`EnhancedCurrentHue` server attribute scenes	True	-
CC.S.A4001	`EnhancedColorMode` server attribute	True	-
CC.S.A4002	`ColorLoopActive` server attribute	True	-
CC.S.A4002.Scene	`ColorLoopActive` server attribute scenes	True	-
CC.S.A4003	`ColorLoopDirection` server attribute	True	-
CC.S.A4003.Scene	`ColorLoopDirection` server attribute scenes	True	-
CC.S.A4004	`ColorLoopTime` server attribute	True	-
CC.S.A4004.Scene	`ColorLoopTime` server attribute scenes	True	-
CC.S.A4005	`ColorLoopStartEnhancedHue` server attribute	True	-
CC.S.A4006	`ColorLoopStoredEnhancedHue` server attribute	True	-
CC.S.A400a	`ColorCapabilities` server attribute	True	-
CC.S.A400b	`ColorTempPhysicalMinMireds` server attribute	True	-
CC.S.A400c	`ColorTempPhysicalMaxMireds` server attribute	True	-
CC.S.A400d	`ColorTempPhysicalMaxMireds` server attribute	True	-
CC.S.A4010	`StartUpColorTemperatureMireds` server attribute	True	-
CC.S.Afffd	`ClusterRevision` server attribute	True	-
CC.S.Afffe	`AttributeReportingStatus` server attribute	False	-
CC.S.C00.Rsp	Move to Hue server command	True	-
CC.S.C01.Rsp	Move Hue server command	True	-
CC.S.C02.Rsp	Step Hue server command	True	-
CC.S.C03.Rsp	Move to Saturation server command	True	-
CC.S.C04.Rsp	Move Saturation server command	True	-
CC.S.C05.Rsp	Step Saturation server command	True	-
CC.S.C06.Rsp	Move to Hue and Saturation server command	True	-

PICS Code	Name	Supported	Notes
CC.S.C07.Rsp	Move to Color server command	True	-
CC.S.C08.Rsp	Move Color server command	True	-
CC.S.C09.Rsp	Step Color server command	True	-
CC.S.C0a.Rsp	Move to Color Temperature server command	True	-
CC.S.C40.Rsp	Enhanced Move to Hue server command	True	-
CC.S.C41.Rsp	Enhanced Move Hue server command	True	-
CC.S.C42.Rsp	Enhanced Step Hue server command	True	-
CC.S.C43.Rsp	Enhanced Move to Hue and Saturation server command	True	-
CC.S.C44.Rsp	Color Loop Set server command	True	-
CC.S.C47.Rsp	Stop Move Step server command	True	-
CC.S.C4b.Rsp	Move Color Temperature server command	True	-
CC.S.C4c.Rsp	Step Color Temperature server command	True	-
CC.C.A0000.Report.Rsp	`CurrentHue` client attribute reports	False	-
CC.C.A0001.Report.Rsp	`CurrentSaturation` client attribute reports	False	-
CC.C.A0003.Report.Rsp	`CurrentX` client attribute reports	False	-
CC.C.A0004.Report.Rsp	`CurrentY` client attribute reports	False	-
CC.C.A0007.Report.Rsp	`ColorTemperatureMireds` client attribute reports	False	-
CC.C.Afffd	`ClusterRevision` client attribute	True	-
CC.C.Afffe	`AttributeReportingStatus` client attribute	False	-
CC.C.C00.Tx	Move to Hue client command	True	-
CC.C.C01.Tx	Move Hue client command	True	-
CC.C.C02.Tx	Step Hue client command	True	-
CC.C.C03.Tx	Move to Saturation client command	True	-
CC.C.C04.Tx	Move Saturation client command	True	-
CC.C.C05.Tx	Step Saturation client command	True	-
CC.C.C06.Tx	Move to Hue and Saturation client command	True	-
CC.C.C07.Tx	Move to Color client command	True	-
CC.C.C08.Tx	Move Color client command	True	-
CC.C.C09.Tx	Step Color client command	True	-
CC.C.C0a.Tx	Move to Color Temperature client command	True	-
CC.C.C40.Tx	Enhanced Move to Hue client command	True	-
CC.C.C41.Tx	Enhanced Move Hue client command	True	-
CC.C.C42.Tx	Enhanced Step Hue client command	True	-
CC.C.C43.Tx	Enhanced Move to Hue and Saturation client command	True	-
CC.C.C44.Tx	Color Loop Set client command	True	-
CC.C.C47.Tx	Stop Move Step client command	True	-
CC.C.C4b.Tx	Move Color Temperature client command	True	-
CC.C.C4c.Tx	Step Color Temperature client command	True	-

5.5.2 Color control cluster functions

5.5.2.1 *ZbZclColorClientAlloc*

```
struct ZbZclClusterT * ZbZclColorClientAlloc(struct ZigBeeT *zb, uint8_t endpoint);
```

Instantiate a new instance of the Color Control client cluster

Table 25. ZbZclColorClientAlloc parameters

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster

Return: Cluster pointer, or NULL if there is an error.

5.5.2.2 *ZbZclColorClientColorLoopSetReq*

```
enum ZclStatusCodeT ZbZclColorClientColorLoopSetReq(struct ZbZclClusterT*clusterPtr, const
struct ZbApsAddrT *dst, struct ZbZclColorClientColorLoopSetReqT*req, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Color Loop Set command

Table 26. ZbZclColorClientColorLoopSetReq parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`req`	Color Loop Set command request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.5.2.3 *ZbZclColorClientMoveColorTempReq*

```
enum ZclStatusCodeT ZbZclColorClientMoveColorTempReq(struct ZbZclClusterT*clusterPtr, const
struct ZbApsAddrT *dst, struct ZbZclColorClientMoveColorTempReqT*req, void (*callback)
(struct ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Move Color Temperature command

Table 27. ZbZclColorClientMoveColorTempReq parameters

Parameter	Description
`clusterPtr`	Cluster instance from which to send this command
`dst`	Requested destination address
`req`	Move Color Temperature command request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.5.2.4 ZbZclColorClientMoveColorXYReq

```
enum ZclStatusCodeT ZbZclColorClientMoveColorXYReq(struct ZbZclClusterT*clusterPtr, const
struct ZbApsAddrT *dst, struct ZbZclColorClientMoveColorXYReqT*req, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Move Color command

Table 28. ZbZclColorClientMoveColorXYReq parameters

Parameter	Description
`clusterPtr`	Cluster instance from which to send this command
`dst`	Requested destination address
`req`	Move Color command request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.5.2.5 ZbZclColorClientMoveHueEnhReq

```
enum ZclStatusCodeT ZbZclColorClientMoveHueEnhReq(struct ZbZclClusterT *clusterPtr, const
struct ZbApsAddrT *dst, struct ZbZclColorClientMoveHueEnhReqT *req, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send an Enhanced Move Hue command

Table 29. ZbZclColorClientMoveHueEnhReq parameters

Parameter	Description
`clusterPtr`	Cluster instance from which to send this command
`dst`	Requested destination address
`req`	Enhanced Move Hue command request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.5.2.6 ZbZclColorClientMoveHueReq

```
enum ZclStatusCodeT ZbZclColorClientMoveHueReq(struct ZbZclClusterT *clusterPtr, const
struct ZbApsAddrT *dst, struct ZbZclColorClientMoveHueReqT *req, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Move Hue command

Table 30. ZbZclColorClientMoveHueReq parameters

Parameter	Description
`clusterPtr`	Cluster instance from which to send this command
`dst`	Requested destination address
`req`	Move Hue command request structure
`callback`	Callback function that is invoked when the response is received

Parameter	Description
arg	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.5.2.7 *ZbZclColorClientMoveSatReq*

```
enum ZclStatusCodeT ZbZclColorClientMoveSatReq(struct ZbZclClusterT *clusterPtr, const
struct ZbApsAddrT *dst, struct ZbZclColorClientMoveSatReqT *req, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Move Saturation command

Table 31. **ZbZclColorClientMoveSatReq parameters**

Parameter	Description
clusterPtr	Cluster instance from which to send this command
dst	Requested destination address
req	Move Saturation command request structure
callback	Callback function that is invoked when the response is received
arg	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.5.2.8 *ZbZclColorClientMoveToColorTempReq*

```
enum ZclStatusCodeT ZbZclColorClientMoveToColorTempReq(struct ZbZclClusterT*clusterPtr,
const struct ZbApsAddrT *dst, struct ZbZclColorClientMoveToColorTempReqT *req, void
(*callback)(struct ZbZclCommandRspT*rsp, void *arg), void *arg);
```

Send a Move to Color Temperature command

Table 32. **ZbZclColorClientMoveToColorTempReq parameters**

Parameter	Description
clusterPtr	Cluster instance from which to send this command
dst	Requested destination address
req	Move to Color Temperature command request structure
callback	Callback function that is invoked when the response is received
arg	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.5.2.9 *ZbZclColorClientMoveToColorXYReq*

```
enum ZclStatusCodeT ZbZclColorClientMoveToColorXYReq(struct ZbZclClusterT*clusterPtr, const
struct ZbApsAddrT *dst, struct ZbZclColorClientMoveToColorXYReqT*req, void (*callback)
(struct ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Move to Color command

Table 33. ZbZclColorClientMoveToColorXYReq Parameters

Parameter	Description
`clusterPtr`	Cluster instance from which to send this command
`dst`	Requested destination address
`req`	Move to Color command request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.5.2.10 *ZbZclColorClientMoveToHueEnhReq*

```
enum ZclStatusCodeT ZbZclColorClientMoveToHueEnhReq(struct ZbZclClusterT*clusterPtr, const
struct ZbApsAddrT *dst, struct ZbZclColorClientMoveToHueEnhReqT*req, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send an Enhanced Move to Hue command

Table 34. ZbZclColorClientMoveToHueEnhReq parameters

Parameter	Description
`clusterPtr`	Cluster instance from which to send this command
`dst`	Requested destination address
`req`	Enhanced Move to Hue command request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.5.2.11 *ZbZclColorClientMoveToHueReq*

```
enum ZclStatusCodeT ZbZclColorClientMoveToHueReq(struct ZbZclClusterT *clusterPtr, const
struct ZbApsAddrT *dst, struct ZbZclColorClientMoveToHueReqT *req, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Move to Hue command

Table 35. ZbZclColorClientMoveToHueReq parameters

Parameter	Description
`clusterPtr`	Cluster instance from which to send this command
`dst`	Requested destination address
`req`	Move to Hue command request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.5.2.12 *ZbZclColorClientMoveToHueSatEnhReq*

```
enum ZclStatusCodeT ZbZclColorClientMoveToHueSatEnhReq(struct ZbZclClusterT*clusterPtr,
const struct ZbApsAddrT *dst, struct ZbZclColorClientMoveToHueSatEnhReqT *req, void
(*callback)(struct ZbZclCommandRspT*rsp, void *arg), void *arg);
```

Send an Enhanced Move to Hue and Saturation command

Table 36. **ZbZclColorClientMoveToHueSatEnhReq parameters**

Parameter	Description
`clusterPtr`	Cluster instance from which to send this command
`dst`	Requested destination address
`req`	Enhanced Move to Hue and Saturation command request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.5.2.13 *ZbZclColorClientMoveToHueSatReq*

```
enum ZclStatusCodeT ZbZclColorClientMoveToHueSatReq(struct ZbZclClusterT*clusterPtr, const
struct ZbApsAddrT *dst, struct ZbZclColorClientMoveToHueSatReqT*req, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Move to Hue and Saturation command

Table 37. **ZbZclColorClientMoveToHueSatReq parameters**

Parameter	Description
`clusterPtr`	Cluster instance from which to send this command
`dst`	Requested destination address
`req`	Move to Hue and Saturation command request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.5.2.14 *ZbZclColorClientMoveToSatReq*

```
enum ZclStatusCodeT ZbZclColorClientMoveToSatReq(struct ZbZclClusterT *clusterPtr, const
struct ZbApsAddrT *dst, struct ZbZclColorClientMoveToSatReqT *req, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Move to Saturation command

Table 38. **ZbZclColorClientMoveToSatReq parameters**

Parameter	Description
`clusterPtr`	Cluster instance from which to send this command
`dst`	Requested destination address
`req`	Move to Saturation command command request structure
`callback`	Callback function that is invoked when the response is received

Parameter	Description
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.5.2.15 *ZbZclColorClientStepColorTempReq*

```
enum ZclStatusCodeT ZbZclColorClientStepColorTempReq(struct ZbZclClusterT*clusterPtr, const
struct ZbApsAddrT *dst, struct ZbZclColorClientStepColorTempReqT*req, void (*callback)
(struct ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Step Color Temperature command

Table 39. **ZbZclColorClientStepColorTempReq parameters**

Parameter	Description
`clusterPtr`	Cluster instance from which to send this command
`dst`	Requested destination address
`req`	Step Color Temperature command request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.5.2.16 *ZbZclColorClientStepColorXYReq*

```
enum ZclStatusCodeT ZbZclColorClientStepColorXYReq(struct ZbZclClusterT*clusterPtr, const
struct ZbApsAddrT *dst, struct ZbZclColorClientStepColorXYReqT*req, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Step Color command

Table 40. **ZbZclColorClientStepColorXYReq parameters**

Parameter	Description
`clusterPtr`	Cluster instance from which to send this command
`dst`	Requested destination address
`req`	Step Color command request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.5.2.17 *ZbZclColorClientStepHueEnhReq*

```
enum ZclStatusCodeT ZbZclColorClientStepHueEnhReq(struct ZbZclClusterT *clusterPtr, const
struct ZbApsAddrT *dst, struct ZbZclColorClientStepHueEnhReqT *req, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send an Enhanced Step Hue command

Table 41. ZbZclColorClientStepHueEnhReq parameters

Parameter	Description
`clusterPtr`	Cluster instance from which to send this command
`dst`	Requested destination address
`req`	Enhanced Step Hue command request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.5.2.18 *ZbZclColorClientStepHueReq*

```
enum ZclStatusCodeT ZbZclColorClientStepHueReq(struct ZbZclClusterT *clusterPtr, const
struct ZbApsAddrT *dst, struct ZbZclColorClientStepHueReqT *req, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Step Hue command

Table 42. ZbZclColorClientStepHueReq parameters

Parameter	Description
`clusterPtr`	Cluster instance from which to send this command
`dst`	Requested destination address
`req`	Step Hue command request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.5.2.19 *ZbZclColorClientStepSatReq*

```
enum ZclStatusCodeT ZbZclColorClientStepSatReq(struct ZbZclClusterT *clusterPtr, const
struct ZbApsAddrT *dst, struct ZbZclColorClientStepSatReqT *req, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Step Saturation command

Table 43. ZbZclColorClientStepSatReq parameters

Parameter	Description
`clusterPtr`	Cluster instance from which to send this command
`dst`	Requested destination address
`req`	Step Saturation command request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.5.2.20 ZbZclColorClientStopMoveStepReq

```
enum ZclStatusCodeT ZbZclColorClientStopMoveStepReq(struct ZbZclClusterT*clusterPtr, const
struct ZbApsAddrT *dst, struct ZbZclColorClientStopMoveStepReqT*req, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Stop Move Step command

Table 44. **ZbZclColorClientStopMoveStepReq parameters**

Parameter	Description
cluster	Cluster instance from which to send this command
dst	Requested destination address
req	Stop Move Step command request structure
callback	Callback function that is invoked when the response is received
arg	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other ZclStatusCodeT value on error.

5.5.2.21 ZbZclColorServerAlloc

```
struct ZbZclClusterT * ZbZclColorServerAlloc(struct ZigBeeT *zb, uint8_t endpoint,
struct ZbZclClusterT *onoff_server, const struct ZbZclAttrT *attribute_list, unsigned int
num_attrs, struct ZbColorClusterConfig *config, void *arg);
```

Instantiate a new instance of the Color Control server cluster

Table 45. **ZbZclColorServerAlloc parameters**

Parameter	Description
zb	Zigbee® stack instance stack instance
endpoint	Endpoint on which to create cluster
onoff_server	OnOff Server cluster pointer for processing commands with the Options fields, may be NULL
attribute_list	List of application defined attributes to be appended, may be NULL
num_attrs	Number of application defined attributes to be added, may be 0 if attribute_list is NULL
config	Configuration containing Color Control capabilities and callbacks for handling requests
arg	Pointer to application data that is provided back to the callback function when invoked

Return: Cluster pointer, or NULL if there is an error.

5.5.3 Color control cluster enumerations

5.5.3.1 ZbZclColorSvrAttrT

Table 46. **Color Control server attribute IDs**

Attribute	Description
ZCL_COLOR_ATTR_CURRENT_HUE	CurrentHue
ZCL_COLOR_ATTR_CURRENT_SAT	CurrentSaturation
ZCL_COLOR_ATTR_REMAINING_TIME	RemainingTime (Optional but is used in mandatory commands)
ZCL_COLOR_ATTR_CURRENT_X	CurrentX

Attribute	Description
ZCL_COLOR_ATTR_CURRENT_Y	CurrentY
ZCL_COLOR_ATTR_DRIFT_COMP	DriftCompensation (optional)
ZCL_COLOR_ATTR_COMPENSATION_TEXT	CompensationText (optional)
ZCL_COLOR_ATTR_COLOR_TEMP_MIREDS	ColorTemperatureMireds
ZCL_COLOR_ATTR_COLOR_MODE	ColorMode
ZCL_COLOR_ATTR_OPTIONS	Options
ZCL_COLOR_ATTR_NUM_PRIMARIES	NumberOfPrimaries
ZCL_COLOR_ATTR_PRIMARY_1X	Primary1X
ZCL_COLOR_ATTR_PRIMARY_1Y	Primary1Y
ZCL_COLOR_ATTR_PRIMARY_1_INTENS	Primary1Intensity
ZCL_COLOR_ATTR_PRIMARY_2X	Primary2X
ZCL_COLOR_ATTR_PRIMARY_2Y	Primary2Y
ZCL_COLOR_ATTR_PRIMARY_2_INTENS	Primary2Intensity
ZCL_COLOR_ATTR_PRIMARY_3X	Primary3X
ZCL_COLOR_ATTR_PRIMARY_3Y	Primary3Y
ZCL_COLOR_ATTR_PRIMARY_3_INTENS	Primary3Intensity
ZCL_COLOR_ATTR_PRIMARY_4X	Primary4X
ZCL_COLOR_ATTR_PRIMARY_4Y	Primary4Y
ZCL_COLOR_ATTR_PRIMARY_4_INTENS	Primary4Intensity
ZCL_COLOR_ATTR_PRIMARY_5X	Primary5X
ZCL_COLOR_ATTR_PRIMARY_5Y	Primary5Y
ZCL_COLOR_ATTR_PRIMARY_5_INTENS	Primary5Intensity
ZCL_COLOR_ATTR_PRIMARY_6X	Primary6X
ZCL_COLOR_ATTR_PRIMARY_6Y	Primary6Y
ZCL_COLOR_ATTR_PRIMARY_6_INTENS	Primary6Intensity
ZCL_COLOR_ATTR_WHITE_POINT_X	WhitePointX (optional)
ZCL_COLOR_ATTR_WHITE_POINT_Y	WhitePointY (optional)
ZCL_COLOR_ATTR_COLOR_POINT_RX	ColorPointRX (optional)
ZCL_COLOR_ATTR_COLOR_POINT_RY	ColorPointRY (optional)
ZCL_COLOR_ATTR_COLOR_POINT_R_INTENS	ColorPointRIntensity (optional)
ZCL_COLOR_ATTR_COLOR_POINT_GX	ColorPointGX (optional)
ZCL_COLOR_ATTR_COLOR_POINT_GY	ColorPointGY (optional)
ZCL_COLOR_ATTR_COLOR_POINT_G_INTENS	ColorPointGIntensity (optional)
ZCL_COLOR_ATTR_COLOR_POINT_BX	ColorPointBX (optional)
ZCL_COLOR_ATTR_COLOR_POINT_BY	ColorPointBY (optional)
ZCL_COLOR_ATTR_COLOR_POINT_B_INTENS	ColorPointBIntensity (optional)
ZCL_COLOR_ATTR_ENH_CURR_HUE	EnhancedCurrentHue
ZCL_COLOR_ATTR_ENH_COLOR_MODE	EnhancedColorMode
ZCL_COLOR_ATTR_COLOR_LOOP_ACTIVE	ColorLoopActive

Attribute	Description
ZCL_COLOR_ATTR_COLOR_LOOP_DIR	`ColorLoopDirection`
ZCL_COLOR_ATTR_COLOR_LOOP_TIME	`ColorLoopTime`
ZCL_COLOR_ATTR_COLOR_LOOP_START_HUE	`ColorLoopStartEnhancedHue`
ZCL_COLOR_ATTR_COLOR_LOOP_STORE_HUE	`ColorLoopStoredEnhancedHue`
ZCL_COLOR_ATTR_COLOR_CAPABILITIES	`ColorCapabilities`
ZCL_COLOR_ATTR_COLOR_TEMP_MIN	`ColorTempPhysicalMinMireds`
ZCL_COLOR_ATTR_COLOR_TEMP_MAX	`ColorTempPhysicalMaxMireds`
ZCL_COLOR_ATTR_COUPLE_COLOR_TL_MIN	`ColorTempPhysicalMaxMireds`
ZCL_COLOR_ATTR_STARTUP_COLOR_TEMP	`StartUpColorTemperatureMireds`

5.5.4 Color control cluster structures

5.5.4.1 *ZbZclColorClientColorLoopSetReqT*

Color Loop Set command structure

Table 47. ZbZclColorClientColorLoopSetReqT parameters

Parameter	Description
`uint8_t update_flags`	Update flags
`uint8_t action`	Action
`uint8_t direction`	Direction
`uint16_t transition_time`	Transition time
`uint16_t start_hue`	Start hue
`uint8_t mask`	Options mask
`uint8_t override`	Options override

5.5.4.2 *ZbZclColorClientMoveColorTempReqT*

Move Color Temperature command structure

Table 48. ZbZclColorClientMoveColorTempReqT parameters

Parameter	Description
`uint8_t move_mode`	Move mode
`uint16_t rate`	Rate
`uint16_t color_temp_min`	Color temperature minimum mireds
`uint16_t color_temp_max`	Color temperature maximum mireds
`uint8_t mask`	Options mask
`uint8_t override`	Options override

5.5.4.3 *ZbZclColorClientMoveColorXYReqT*

Move Color command structure

Table 49. ZbZclColorClientMoveColorXYReqT parameters

Parameter	Description
`uint16_t rate_x`	RateX
`uint16_t rate_y`	RateY
`uint8_t mask`	Options mask
`uint8_t override`	Options override

5.5.4.4 *ZbZclColorClientMoveHueEnhReqT*

Enhanced Move Hue command structure

Table 50. ; ZbZclColorClientMoveHueEnhReqT parameters

Parameter	Desxcription
`uint8_t move_mode`	Move mode
`uint16_t rate`	Rate
`uint8_t mask`	Options mask
`uint8_t override`	Options override

5.5.4.5 *ZbZclColorClientMoveHueReqT*

Move Hue command structure

Table 51. ZbZclColorClientMoveHueReqT Parameters

Parameter	Description
`uint8_t move_mode`	Move mode
`uint8_t rate`	Rate
`uint8_t mask`	Options mask
`uint8_t override`	Options override

5.5.4.6 *ZbZclColorClientMoveSatReqT*

Move Saturation command structure

Table 52. ZbZclColorClientMoveSatReqT parameters

Parameter	Descption
`uint8_t move_mode`	Move mode
`uint8_t rate`	Rate
`uint8_t mask`	Options mask
`uint8_t override`	Options override

5.5.4.7 *ZbZclColorClientMoveToColorTempReqT*

Move to Color Temperature command structure

Table 53. ZbZclColorClientMoveToColorTempReqT parameters

Parameter	Description
`uint16_t color_temp`	Color temperature mireds

Parameter	Description
`uint16_t transition_time`	Transition time
`uint8_t mask`	Options mask
`uint8_t override`	Options override

5.5.4.8 *ZbZclColorClientMoveToColorXYReqT*

Move to Color command structure

Table 54. **ZbZclColorClientMoveToColorXYReqT parameters**

Parameter	Description
`uint16_t color_x`	`ColorX`
`uint16_t color_y`	`ColorY`
`uint16_t transition_time`	Transition time
`uint8_t mask`	Options mask
`uint8_t override`	Options override

5.5.4.9 *ZbZclColorClientMoveToHueEnhReqT*

Enhanced Move to Hue command structure

Table 55. **ZbZclColorClientMoveToHueEnhReqT parameters**

Parameter	Description
`uint16_t enh_hue`	Enhanced hue
`uint8_t direction`	Direction
`uint16_t transition_time`	Transition time
`uint8_t mask`	Options mask
`uint8_t override`	`OptionsOverride`

5.5.4.10 *ZbZclColorClientMoveToHueReqT*

Move to Hue command structure

Table 56. **ZbZclColorClientMoveToHueReqT parameters**

Parameter	Description
`uint8_t hue`	Hue
`uint8_t direction`	Direction
`uint16_t transition_time`	Transition time
`uint8_t mask`	Options mask
`uint8_t override`	Options override

5.5.4.11 *ZbZclColorClientMoveToHueSatEnhReqT*

Enhanced Move to Hue and Saturation command structure

Table 57. ZbZclColorClientMoveToHueSatEnhReqT parameters

Parameter	Description
`uint16_t enh_hue`	Enhanced hue
`uint8_t sat`	Saturation
`uint16_t transition_time`	Transition time
`uint8_t mask`	Options mask
`uint8_t override`	Options override

5.5.4.12 *ZbZclColorClientMoveToHueSatReqT*

Move to Hue and Saturation command structure

Table 58. ZbZclColorClientMoveToHueSatReqT parameters

Parameter	Description
`uint8_t hue`	Hue
`uint8_t sat`	Saturation
`uint16_t transition_time`	Transition time
`uint8_t mask`	Options mask
`uint8_t override`	Options override

5.5.4.13 *ZbZclColorClientMoveToSatReqT*

Move to Saturation command structure

Table 59. ZbZclColorClientMoveToSatReqT parameters

Parameter	Description
`uint8_t sat`	Saturation
`uint16_t transition_time`	Transition time
`uint8_t mask`	Options mask
`uint8_t override`	Options override

5.5.4.14 *ZbZclColorClientStepColorTempReqT*

Step Color Temperature command structure

Table 60. ZbZclColorClientStepColorTempReqT parameters

Parameter	Description
`uint8_t step_mode`	Step mode
`uint16_t step_size`	Step size
`uint16_t transition_time`	Transition time
`uint16_t color_temp_min`	Color temperature minimum mireds
`uint16_t color_temp_max`	Color temperature maximum mireds
`uint8_t mask`	Options mask
`uint8_t override`	Options override

5.5.4.15 *ZbZclColorClientStepColorXYReqT*

Step Color command structure

Table 61. parameters

Parameter	Description
`uint16_t step_x`	`StepX`
`uint16_t step_y`	`StepY`
`uint16_t transition_time`	Transition time
`uint8_t mask`	Options mask
`uint8_t override`	`OptionsOverride`

5.5.4.16 *ZbZclColorClientStepHueEnhReqT*

Enhanced Step Hue command structure

Table 62. ZbZclColorClientStepHueEnhReqT parameters

Parameter	Description
`uint8_t step_mode`	Step mode
`uint16_t step_size`	Step size
`uint16_t transition_time`	Transition time
`uint8_t mask`	Options mask
`uint8_t override`	Options override

5.5.4.17 *ZbZclColorClientStepHueReqT*

Step Hue command structure

Table 63. ZbZclColorClientStepHueReqT parameters

Parameter	Description
`uint8_t step_mode`	Step mode
`uint8_t step_size`	Step size
`uint8_t transition_time`	Transition time
`uint8_t mask`	Options mask
`uint8_t override`	Options override

5.5.4.18 *ZbZclColorClientStepSatReqT*

Step Saturation command structure

Table 64. ZbZclColorClientStepSatReqT parameters

Parameter	Description
`uint8_t step_mode`	Step mode
`uint8_t step_size`	Step size
`uint8_t transition_time`	Transition time
`uint8_t mask`	Options mask
`uint8_t override`	Options override

5.5.4.19 *ZbZclColorClientStopMoveStepReqT*

Stop Move Step command structure

Table 65. ZbZclColorClientStopMoveStepReqT Parameters

Parameter	Description
`uint8_t mask`	Options mask
`uint8_t override`	Options override

5.5.4.20 *ZbZclColorServerCallbacksT*

Color Control Server callbacks configuration

Table 66. ZbZclColorServerCallbacksT Parameters

Parameter	Decription
`move_to_hue` (callback function pointer)	`enum ZclStatusCodeT (*move_to_hue)(struct ZbZclClusterT*cluster, struct ZbZclColorClientMoveToHueReqT *req, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of Move to Hue command. The application is expected to update ZCL_COLOR_ATTR_CURRENT_HUE, and ZCL_COLOR_ATTR_REMAINING_TIME
`move_hue` (callback function pointer)	`enum ZclStatusCodeT (*move_hue)(struct ZbZclClusterT*cluster, struct ZbZclColorClientMoveHueReqT *req, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of Move Hue command. The application is expected to update ZCL_COLOR_ATTR_CURRENT_HUE, and ZCL_COLOR_ATTR_REMAINING_TIME
`step_hue` (callback function pointer)	`enum ZclStatusCodeT (*step_hue)(struct ZbZclClusterT*cluster, struct ZbZclColorClientStepHueReqT *req, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of Step Hue command. The application is expected to update ZCL_COLOR_ATTR_CURRENT_HUE, and ZCL_COLOR_ATTR_REMAINING_TIME
`move_to_sat` (callback function pointer)	`enum ZclStatusCodeT (*move_to_sat)(struct ZbZclClusterT*cluster, struct ZbZclColorClientMoveToSatReqT *req, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of Move to Saturation command. The application is expected to update ZCL_COLOR_ATTR_CURRENT_SAT, and ZCL_COLOR_ATTR_REMAINING_TIME
`move_sat` (callback function pointer)	`enum ZclStatusCodeT (*move_sat)(struct ZbZclClusterT*cluster, struct ZbZclColorClientMoveSatReqT *req, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of Move Saturation command. The application is expected to update ZCL_COLOR_ATTR_CURRENT_SAT, and ZCL_COLOR_ATTR_REMAINING_TIME
step_sat (callback function pointer)	`enum ZclStatusCodeT (*step_sat)(struct ZbZclClusterT*cluster, struct ZbZclColorClientStepSatReqT *req, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of Step Saturation command. The application is expected to update ZCL_COLOR_ATTR_CURRENT_SAT, and ZCL_COLOR_ATTR_REMAINING_TIME

Parameter	Decription
`move_to_hue_sat` (callback function pointer)	`enum ZclStatusCodeT (*move_to_hue_sat)(struct ZbZclClusterT *cluster, struct ZbZclColorClientMoveToHueSatReqT *req, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of Move to Hue and Saturation command. The application is expected to update ZCL_COLOR_ATTR_CURRENT_HUE, ZCL_COLOR_ATTR_CURRENT_SAT, and ZCL_COLOR_ATTR_REMAINING_TIME
`move_to_color_xy` (callback function pointer)	`enum ZclStatusCodeT (*move_to_color_xy)(struct ZbZclClusterT*cluster,struct ZbZclColorClientMoveToColorXYReqT *req, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of Move to Color command. The application is expected to update ZCL_COLOR_ATTR_CURRENT_X, ZCL_COLOR_ATTR_CURRENT_Y, and ZCL_COLOR_ATTR_REMAINING_TIME
`move_color_xy` (callback function pointer)	`enum ZclStatusCodeT (*move_color_xy)(struct ZbZclClusterT *cluster,struct ZbZclColorClientMoveColorXYReqT *req, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of Move Color command. The application is expected to update ZCL_COLOR_ATTR_CURRENT_X, ZCL_COLOR_ATTR_CURRENT_Y, and ZCL_COLOR_ATTR_REMAINING_TIME
`step_color_xy` (callback function pointer)	`enum ZclStatusCodeT (*step_color_xy)(struct ZbZclClusterT *cluster,struct ZbZclColorClientStepColorXYReqT *req, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of Step Color command. The application is expected to update ZCL_COLOR_ATTR_CURRENT_X, ZCL_COLOR_ATTR_CURRENT_Y, and ZCL_COLOR_ATTR_REMAINING_TIME
`move_to_color_temp` (callback function pointer)	`enum ZclStatusCodeT (*move_to_color_temp)(struct ZbZclClusterT *cluster, struct ZbZclColorClientMoveToColorTempReqT *req, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of Move to Color Temperature command. The application is expected to update ZCL_COLOR_ATTR_COLOR_TEMP_MIREDS, and ZCL_COLOR_ATTR_REMAINING_TIME
`move_to_hue_enh` (callback function pointer)	`enum ZclStatusCodeT (*move_to_hue_enh)(struct ZbZclClusterT *cluster,struct ZbZclColorClientMoveToHueEnhReqT *req, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of Enhanced Move to Hue command. The application is expected to update ZCL_COLOR_ATTR_ENH_CURR_HUE, and ZCL_COLOR_ATTR_REMAINING_TIME
`move_hue_enh` (callback function pointer)	`enum ZclStatusCodeT (*move_hue_enh)(struct ZbZclClusterT*cluster, struct ZbZclColorClientMoveHueEnhReqT *req, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of Move Hue command. The application is expected to update ZCL_COLOR_ATTR_ENH_CURR_HUE, and ZCL_COLOR_ATTR_REMAINING_TIME
`step_hue_enh` (callback function pointer)	`enum ZclStatusCodeT (*step_hue_enh)(struct ZbZclClusterT*cluster, struct ZbZclColorClientStepHueEnhReqT *req, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of Enhanced Step Hue command. The application is expected to update ZCL_COLOR_ATTR_ENH_CURR_HUE, and ZCL_COLOR_ATTR_REMAINING_TIME
`move_to_hue_sat_enh` (callback function pointer)	`enum ZclStatusCodeT (*move_to_hue_sat_enh)(struct ZbZclClusterT *cluster, struct ZbZclColorClientMoveToHueSatEnhReqT *req, struct ZbZclAddrInfoT *srcInfo, void *arg)`

Parameter	Decription
	Callback to application, invoked on receipt of Enhanced Move to Hue and Saturation command. Theapplication is expected to update ZCL_COLOR_ATTR_ENH_CURR_HUE, ZCL_COLOR_ATTR_CURRENT_SAT, and ZCL_COLOR_ATTR_REMAINING_TIME
`color_loop_set` (callback function pointer)	`enum ZclStatusCodeT (*color_loop_set)(struct ZbZclClusterT *cluster, struct ZbZclColorClientColorLoopSetReqT *req, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of Color Loop Set command. The application is expected to update the following attributes according to the update flags and action fields: • ZCL_COLOR_ATTR_COLOR_LOOP_ACTIVE • ZCL_COLOR_ATTR_COLOR_LOOP_DIR • ZCL_COLOR_ATTR_COLOR_LOOP_TIME • ZCL_COLOR_ATTR_COLOR_LOOP_START_HUE • ZCL_COLOR_ATTR_ENH_CURR_HUE • ZCL_COLOR_ATTR_COLOR_LOOP_STORE_HUE • ZCL_COLOR_ATTR_REMAINING_TIME
`stop_move_step` (callback function pointer)	`enum ZclStatusCodeT (*stop_move_step)(struct ZbZclClusterT *cluster,struct ZbZclColorClientStopMoveStepReqT *req, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of Stop Move Step command. The application is expected to update ZCL_COLOR_ATTR_REMAINING_TIME
`move_color_temp` (callback function pointer)	`enum ZclStatusCodeT (*move_color_temp)(struct ZbZclClusterT *cluster, struct ZbZclColorClientMoveColorTempReqT *req, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of Move Color Temperature command. The application is expected to update ZCL_COLOR_ATTR_COLOR_TEMP_MIREDS, and ZCL_COLOR_ATTR_REMAINING_TIME
`step_color_temp` (callback function pointer)	`enum ZclStatusCodeT (*step_color_temp)(struct ZbZclClusterT *cluster,struct ZbZclColorClientStepColorTempReqT *req, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of Step Color Temperature command. The application is expected to update ZCL_COLOR_ATTR_COLOR_TEMP_MIREDS, and ZCL_COLOR_ATTR_REMAINING_TIME

5.6 Commissioning cluster

```
#include "zcl/general/zcl.commission.h"
```

5.6.1 Commissioning cluster description

Table 67. **Commissioning cluster PICS code description**

PICS Code	Name	Supported	Notes
CS.S	Server	True	-
CS.C	Client	True	-
CS.S.A0000	`ShortAddress` server attribute	True	-
CS.S.A0001	`ExtendedPANId` server attribute	True	-
CS.S.A0002	`PANId` server attribute	True	-
CS.S.A0003	`Channelmask` server attribute	True	-
CS.S.A0004	`ProtocolVersion` server attribute	True	-
CS.S.A0005	`StackProfile` server attribute	True	-

PICS Code	Name	Supported	Notes
CS.S.A0006	`StartupControl` server attribute	True	-
CS.S.A0010	`TrustCenterAddress` server attribute	True	-
CS.S.A0012	`NetworkKey` server attribute	True	-
CS.S.A0013	`UseInsecureJoin` server attribute	True	-
CS.S.A0014	`PreconfiguredLinkKey` server attribute	True	-
CS.S.A0015	`NetworkKeySeqNum` server attribute	True	-
CS.S.A0016	`NetworkKeyType` server attribute	True	-
CS.S.A0017	`NetworkManagerAddress` server attribute	True	-
CS.S.A0020	`ScanAttempts` server attribute	True	Optional
CS.S.Afffd	`ClusterRevision` server attribute	True	-
CS.S.Afffe	`AttributeReportingStatus` server attribute	False	-
CS.S.C00.Rsp	Restart Device server command	Depends on customer application	-
CS.S.C01.Rsp	Save Startup Parameters server command	Depends on customer application	-
CS.S.C02.Rsp	Restore Startup Parameters server command	Depends on customer application	-
CS.S.C03.Rsp	Reset Startup Parameters server command	Depends on customer application	-
CS.S.C00.Tx	Restart Device Response server command	Depends on customer application	-
CS.S.C01.Tx	Save Startup Parameters Response server command	Depends on customer application	-
CS.S.C02.Tx	Restore Startup Parameters Response server command	Depends on customer application	-
CS.S.C03.Tx	Reset Startup Parameters Response server command	Depends on customer application	-
CS.C.Afffd	`ClusterRevision` client attribute	True	-
CS.C.Afffe	`AttributeReportingStatus` client attribute	False	-
CS.C.C00.Rsp	Restart Device Response client command	Depends on customer application	-
CS.C.C01.Rsp	Save Startup Parameters client command	Depends on customer application	-
CS.C.C02.Rsp	Restore Startup Parameters Response client command	Depends on customer application	-
CS.C.C03.Rsp	Reset Startup Parameters Response client command	Depends on customer application	-
CS.C.C00.Tx	Restart Device client command	True	-
CS.C.C01.Tx	Save Startup Parameters client command	True	-
CS.C.C02.Tx	Restore Startup Parameters client command	True	-
CS.C.C03.Tx	Reset Startup Parameters client command	True	-

5.6.2 Commissioning cluster functions

5.6.2.1 *ZbZclCommissionClientAlloc*

```
struct ZbZclClusterT * ZbZclCommissionClientAlloc(struct ZigBeeT *zb, uint8_t endpoint,
uint16_t profile, bool aps_secured);
```

Create a new instance of the Commissioning Client cluster

Table 68. **ZbZclCommissionClientAlloc parameters**

Parameter	Description
`zb`	Zigbee® stack instance

Parameter	Description
endpoint	Set to ZB_ENDPOINT_BCAST if using Inter-PAN for communicating ZCL messages. Otherwise, set a valid ZCL endpoint.
profile	Profile ID for this cluster (e.g. ZCL_PROFILE_HOME_AUTOMATION)
aps_secured	APS security - true if APS security enabled, else false

Return: Cluster pointer, or NULL if there is an error.

5.6.2.2 *ZbZclCommissionClientEnable*

```
enum ZclStatusCodeT ZbZclCommissionClientEnable(struct ZbZclClusterT *cluster, struct
ZbZclCommissionClientEnableInfoT *info);
```

Enable Commissioning Client by configuring MAC layer to listen for packets.

Table 69. ZbZclCommissionClientEnable parameters

Parameter	Description
cluster	Cluster instance from which to send this command
info	Commissioning Client Enable Information structure

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.6.2.3 *ZbZclCommissionClientSendResetStartup*

```
enum ZclStatusCodeT ZbZclCommissionClientSendResetStartup(struct ZbZclClusterT*cluster,
uint64_t dst_ext, uint8_t dst_ep, struct ZbZclCommissionClientResetStartup *req, void
(*callback)(struct ZbZclCommandRspT*rsp, void *arg), void *arg);
```

Send a Reset Startup Parameters command

Table 70. ZbZclCommissionClientSendResetStartup parameters

Parameter	Description
cluster	Cluster instance from which to send this command
dst_ext	Extended address of the device to send this command
dst_ep	If cluster is not using Inter-PAN, this is the destination endpoint to send the command to.
req	Reset Startup Parameters command structure
callback	Callback function that is invoked when the response is received
arg	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.6.2.4 *ZbZclCommissionClientSendRestart*

```
enum ZclStatusCodeT ZbZclCommissionClientSendRestart(struct ZbZclClusterT *cluster, uint64_t
dst_ext, uint8_t dst_ep, struct ZbZclCommissionClientRestartDev *req, void (*callback)
(struct ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Restart Device command

Table 71. ZbZclCommissionClientSendRestart parameters

Parameter	Description
cluster	Cluster instance from which to send this command
dst_ext	Extended address of the device to send this command
dst_ep	If cluster is not using Inter-PAN, this is the destination endpoint to send the command to.
req	Restart Device command structure
callback	Callback function that is invoked when the response is received
arg	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other ZclStatusCodeT value on error.

5.6.2.5 *ZbZclCommissionClientSendRestoreStartup*

```
enum ZclStatusCodeT ZbZclCommissionClientSendRestoreStartup(struct ZbZclClusterT*cluster,
uint64_t dst_ext, uint8_t dst_ep, struct ZbZclCommissionClientRestoreStartup *req, void
(*callback)(struct ZbZclCommandRspT*rsp, void *arg), void *arg);
```

Send a Restore Startup Parameters command

Table 72. ZbZclCommissionClientSendRestoreStartup parameters

Parameter	Description
cluster	Cluster instance from which to send this command
dst_ext	Extended address of the device to send this command
dst_ep	If cluster is not using Inter-PAN, this is the destination endpoint to send the command to.
req	Restore Startup Parameters command structure
callback	Callback function that is invoked when the response is received
arg	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other ZclStatusCodeT value on error.

5.6.2.6 *ZbZclCommissionServerAlloc*

```
struct ZbZclClusterT * ZbZclCommissionServerAlloc(struct ZigBeeT *zb, uint8_t endpoint,
uint16_t profile, bool aps_secured, struct ZbZclCommissionServerCallbacksT *callbacks, void
*arg);
```

Create a new instance of the Commissioning Server cluster

Table 73. ZbZclCommissionServerAlloc parameters

Parameter	Description
zb	Zigbee® stack instance
endpoint	Set to ZB_ENDPOINT_BCAST if using Inter-PAN for communicating ZCL messages. Otherwise, set a valid ZCL endpoint.
profile	Profile ID for this cluster (e.g. ZCL_PROFILE_HOME_AUTOMATION)
aps_secured	APS security - true if APS security enabled, else false
callback	Structure containing any callback function pointers for this cluster
arg	Pointer to application data that is provided back to the callback function when invoked

Return: Cluster pointer, or NULL if there is an error.

5.6.2.7 *ZbZclCommissionServerEnable*

```
enum ZclStatusCodeT ZbZclCommissionServerEnable(struct ZbZclClusterT *cluster, bool enable,
struct ZbZclCommissionServerEnableInfoT *info);
```

Enable the Commissioning Server by configuring the MAC layer to listen for packets. If enable is false, then Commissioning Server stops processing any received Commissioning packets.

Table 74. ZbZclCommissionServerEnable parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`enable`	Enable or disable the ability to receive and process Commissioning commands
`info`	Commissioning Server Enable Information structure. Contains information needed to start listening for Commissioning commands on a given channel. Optional and only applicable if enable is true. This may be NULL if already configured and operational on a network.

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.6.2.8 *ZbZclCommissionServerGetStartup*

```
enum ZclStatusCodeT ZbZclCommissionServerGetStartup(struct ZbZclClusterT *cluster, struct
ZbStartupT *config);
```

Load startup configuration from Cluster Server's attributes to the stack's ZbStartupT structure

Table 75. ZbZclCommissionServerGetStartup parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`config`	Zigbee® Stack Startup Configuration structure

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.6.2.9 *ZbZclCommissionServerResetStartup*

```
enum ZclStatusCodeT ZbZclCommissionServerResetStartup(struct ZbZclClusterT*cluster);
```

Reset startup configurations cluster attributes back to defaults

Table 76. ZbZclCommissionServerResetStartup parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.6.2.10 *ZbZclCommissionServerSendResetStartupRsp*

```
enum ZclStatusCodeT ZbZclCommissionServerSendResetStartupRsp(struct ZbZclClusterT*cluster,
struct ZbZclAddrInfoT *dst, struct ZbZclCommissionServerResetStartupRsp*rsp);
```

Send a Reset Startup Parameters Response command

Table 77. parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Destination address for response, including sequence number and tx options
`rsp`	Reset Startup Parameters Response command structure

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.6.2.11 *ZbZclCommissionServerSendRestartRsp*

```
enum ZclStatusCodeT ZbZclCommissionServerSendRestartRsp(struct ZbZclClusterT*cluster, struct
ZbZclAddrInfoT *dst, struct ZbZclCommissionServerRestartDevRsp*rsp);
```

Send a Restart Device Response command

Table 78. ZbZclCommissionServerSendRestartRsp parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Destination address for response, including sequence number and tx options
`rsp`	Restart Device Response command structure

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.6.2.12 *ZbZclCommissionServerSendRestoreStartupRsp*

```
enum ZclStatusCodeT ZbZclCommissionServerSendRestoreStartupRsp(struct ZbZclClusterT*cluster,
struct ZbZclAddrInfoT *dst, struct ZbZclCommissionServerRestoreStartupRsp*rsp);
```

Send a Restore Startup Parameters Response command

Table 79. ZbZclCommissionServerSendRestoreStartupRsp Parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Destination address for response, including sequence number and tx options
`rsp`	Restore Startup Parameters Response command structure

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.6.2.13 *ZbZclCommissionServerSendSaveStartupRsp*

```
enum ZclStatusCodeT ZbZclCommissionServerSendSaveStartupRsp(struct ZbZclClusterT*cluster,
struct ZbZclAddrInfoT *dst, struct ZbZclCommissionServerSaveStartupRsp*rsp);
```

Send a Save Startup Parameters Response command

Table 80. ZbZclCommissionServerSendSaveStartupRsp Parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Destination address for response, including sequence number and tx options
`rsp`	Save Startup Parameters Response command structure

Return: ZCL_STATUS_SUCCESS if successful, or other ZclStatusCodeT value on error.

5.6.3 Commissioning cluster enumerations

5.6.3.1 *ZbZclCommissionServerAttrT*

Table 81. ZbZclCommissionServerAttrT Commissioning server attribute IDs

Attribute	Description
ZCL_COMMISSION_SVR_ATTR_SHORT_ADDR	ShortAddress. This must only be set if StartupControl is equal to 0x00 (ZbStartTypePreconfigured) or 0x02 (ZbStartTypeRejoin).
ZCL_COMMISSION_SVR_ATTR_EPID	ExtendedPANId
ZCL_COMMISSION_SVR_ATTR_PANID	PANId. This must only be set if StartupControl is equal to 0x00 (ZbStartTypePreconfigured) or 0x02 (ZbStartTypeRejoin).
ZCL_COMMISSION_SVR_ATTR_CHANNELMASK	Channelmask
ZCL_COMMISSION_SVR_ATTR_PROTOCOLVER	ProtocolVersion. Default value is 0x0002 (ZB_PROTOCOL_VERSION_2007)
ZCL_COMMISSION_SVR_ATTR_STACKPROFILE	StackProfile. Default value is 0x02 (ZB_NWK_STACK_PROFILE_PRO)
ZCL_COMMISSION_SVR_ATTR_STARTUPCONTROL	StartupControl: • 0x00 = ZbStartTypePreconfigured, • 0x01 = ZbStartTypeForm, • 0x02 = ZbStartTypeRejoin, • 0x03 = ZbStartTypeJoin.
ZCL_COMMISSION_SVR_ATTR_TCADDR	TrustCenterAddress. This should only be set if StartupControl is equal to 0x00 (ZbStartTypePreconfigured). Otherwise it should be zero to allow the Transport Key to be decrypted and processed correctly during joining.
ZCL_COMMISSION_SVR_ATTR_TCMASTER	TrustCenterMasterKey (optional)
ZCL_COMMISSION_SVR_ATTR_NWKKEY	NetworkKey. This must only be set if StartupControl is equal to 0x00 (ZbStartTypePreconfigured).
ZCL_COMMISSION_SVR_ATTR_USEINSECJOIN	UseInsecureJoin
ZCL_COMMISSION_SVR_ATTR_PRECONFLINKKEY	PreconfiguredLinkKey
ZCL_COMMISSION_SVR_ATTR_NWKKEYSEQNUM	NetworkKeySeqNum. This must only be set if StartupControl is equal to 0x00 (ZbStartTypePreconfigured).
ZCL_COMMISSION_SVR_ATTR_NWKKEYTYPE	NetworkKeyType. This must only be set if StartupControl is equal to 0x00 (ZbStartTypePreconfigured).
ZCL_COMMISSION_SVR_ATTR_NWKMGRADDR	NetworkManagerAddress. This must only be set if StartupControl is equal to 0x00 (ZbStartTypePreconfigured) or 0x02 (ZbStartTypeRejoin).
ZCL_COMMISSION_SVR_ATTR_SCANATTEMPTS	ScanAttempts (optional)
ZCL_COMMISSION_SVR_ATTR_TIMEBTWSCANS	TimeBetweenScans (optional)

Attribute	Description
ZCL_COMMISSION_SVR_ATTR_REJOININTERVAL	`RejoinInterval` (optional)
ZCL_COMMISSION_SVR_ATTR_MAXREJOININTERV AL	`MaxRejoinInterval` (optional)
ZCL_COMMISSION_SVR_ATTR_POLLRATE	`IndirectPollRate` (optional)
ZCL_COMMISSION_SVR_ATTR_PARENTRETRYTHR ESH	`ParentRetryThreshold` (optional)
ZCL_COMMISSION_SVR_ATTR_CONCFLAG	`ConcentratorFlag` (optional)
ZCL_COMMISSION_SVR_ATTR_CONCRADIUS	`ConcentratorRadius` (optional)
ZCL_COMMISSION_SVR_ATTR_CONCDISCTIME	`ConcentratorDiscoveryTime` (optional)

5.6.4 Commissioning cluster structures

5.6.4.1 *ZbZclCommissionClientEnableInfoT*

Commissioning Client Enable Information structure

Table 82. **ZbZclCommissionClientEnableInfoT Parameters**

Parameter	Description
`uint8_t page`	Page
`uint8_t channel`	Channel

5.6.4.2 *ZbZclCommissionClientResetStartup*

Reset Startup Parameters command structure

Table 83. **ZbZclCommissionClientResetStartup parameters**

Parameter	Description
`uint8_t options`	Options, for example ZCL_COMMISS_RESET_OPTS_RESET_CURR
`uint8_t index`	Index

5.6.4.3 *ZbZclCommissionClientRestartDev*

Restart Device command structure

Table 84. **ZbZclCommissionClientRestartDev parameters**

Parameter	Description
`uint8_t options`	Options, for example ZCL_COMMISS_RESTART_OPTS_MODE_MASK
`uint8_t delay`	Delay (seconds)
`uint8_t jitter`	Jitter - RAND(jitter * 80) milliseconds

5.6.4.4 *ZbZclCommissionClientRestoreStartup*

Restore Startup Parameters command structure

Table 85. **ZbZclCommissionClientRestoreStartup parameters**

Parameter	Description
`uint8_t options`	Options (Reserved)
`uint8_t index`	Index

5.6.4.5 *ZbZclCommissionClientSaveStartup*

Save Startup Parameters command structure

Table 86. **ZbZclCommissionClientSaveStartup parameters**

Parmater	Description
`uint8_t options`	Options (Reserved)
`uint8_t index`	Index

5.6.4.6 *ZbZclCommissionServerCallbacksT*

Commissioning Server callbacks configuration

Table 87. **ZbZclCommissionServerCallbacksT parameters**

Paramter	Description
`restart_device` (callback function pointer)	`enum ZclStatusCodeT (*restart_device)(struct ZbZclClusterT*cluster,struct ZbZclCommissionClientRestartDev *req, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of Restart Device command. Must call `ZbZclCommissionServerSendRestartRsp` to send a response. Since the application calls `ZbStartup` or similar, the application must wait and let the stack send the response before something like `ZbStartup` is called. 100 milliseconds is be sufficient (ZCL_COMMISSION_RESTART_DEVICE_DELAY_MS).
`save_startup` (callback function pointer)	`enum ZclStatusCodeT (*save_startup)(struct ZbZclClusterT*cluster,structZbZclCommissionClientSaveStartup*req, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of Save Startup Parameters command.
`restore_startup` (callback function pointer)	`EnumZclStatusCodeT(*restore_startup)(struct ZbZclClusterT*cluster,struct ZbZclCommissionClientRestoreStartup*req,struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of Restore Startup Parameters command.
`reset_startup` (callback function pointer)	`EnumZclStatusCodeT(*reset_startup)(struct ZbZclClusterT*cluster,struct ZbZclCommissionClientResetStartup*req,struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of Reset Startup Parameters command.

5.6.4.7 *ZbZclCommissionServerEnableInfoT*

Commissioning Server Enable Information structure

Table 88. **ZbZclCommissionServerEnableInfoT parameters**

Parameter	Description
`uint8_t page`	Page
`uint8_t channel`	Channel

5.6.4.8 *ZbZclCommissionServerResetStartupRsp*

Reset Startup Parameters Response command structure

Table 89. ZbZclCommissionServerResetStartupRsp parameters

Paramter	Description
`enum ZclStatusCodeT status`	ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error

5.6.4.9 *ZbZclCommissionServerRestartDevRsp*

Restart Device Response command structure

Table 90. ZbZclCommissionServerRestartDevRsp parameters

Parameter	Description
`enum ZclStatusCodeT status`	ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error

5.6.4.10 *ZbZclCommissionServerRestoreStartupRsp*

Restore Startup Parameters Response command structure

Table 91. ZbZclCommissionServerRestoreStartupRsp parameters

Parameter	Descripton
`enum ZclStatusCodeT status`	ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error

5.6.4.11 *ZbZclCommissionServerSaveStartupRsp*

Save Startup Parameters Response command structure

Table 92. ZbZclCommissionServerSaveStartupRsp parameters

Parameter	Description
`enum ZclStatusCodeT status`	ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error

5.7 Dehumidification Control cluster

```
#include "zcl/general/zcl.dehum.ctrl.h"
```

5.7.1 Dehumidification Control cluster description

Table 93. Dehumidification Control cluster PICS code description

PICS Code	Name	Supported	Notes
DHUM.S	Server	True	-
DHUM.C	Client	True	-
DHUM.S.A0001	`DehumidificationCooling` server attribute	True	-
DHUM.S.A0001.Report.Tx	`DehumidificationCooling` server attribute reports	True	-
DHUM.S.A0010	`RHDehumidificationSetpoint` server attribute	True	-
DHUM.S.A0013	`DehumidificationHysteresis` server attribute	True	-
DHUM.S.A0014	`DehumidificationMaxCool` server attribute	True	-
DHUM.S.Afffd	`ClusterRevision` server attribute	True	-
DHUM.S.Afffe	`AttributeReportingStatus` server attribute	False	-
DHUM.C.A0001.Report.Tx	`DehumidificationCooling` client attribute reports	False	-
DHUM.C.Afffd	`ClusterRevision` client attribute	True	-

PICS Code	Name	Supported	Notes
DHUM.C.Affe	`AttributeReportingStatus` client attribute	False	-

5.7.2 Dehumidification Control cluster functions

5.7.2.1 *ZbZclDehumCtrlClientAlloc*

```
struct ZbZclClusterT * ZbZclDehumCtrlClientAlloc(struct ZigBeeT *zb, uint8_t endpoint);
```

Instantiate a new instance of the Dehumidification Control client cluster

Table 94. **ZbZclDehumCtrlClientAlloc parameters**

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster

Return: Cluster pointer, or NULL if there is an error.

5.7.2.2 *ZbZclDehumCtrlServerAlloc*

```
struct ZbZclClusterT * ZbZclDehumCtrlServerAlloc(struct ZigBeeT *zb, uint8_t
endpoint);
```

Instantiate a new instance of the Dehumidification Control server cluster

Table 95. **ZbZclDehumCtrlServerAlloc parameters**

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster

Return: Cluster pointer, or NULL if there is an error.

5.7.3 Dehumidification Control cluster enumerations

5.7.3.1 *ZbZclDehumCtrlServerAttrT*

Table 96. **Dehumidification Control Attribute Ids**

Attributes	Description
ZCL_DEHUM_CTRL_SVR_ATTR_REL_HUM	Relative humidity
ZCL_DEHUM_CTRL_SVR_ATTR_DEHUM_COOLING	Dehumidification cooling
ZCL_DEHUM_CTRL_SVR_ATTR_RHDH_SETPT	RHDehumidification setpoint
ZCL_DEHUM_CTRL_SVR_ATTR_RH_MODE	Relative humidity mode
ZCL_DEHUM_CTRL_SVR_ATTR_DH_LOCKOUT	Dehumidification lockout
ZCL_DEHUM_CTRL_SVR_ATTR_DH_HYS	Dehumidification hysteresis
ZCL_DEHUM_CTRL_SVR_ATTR_DH_MAX_COOL	Dehumidification max cool
ZCL_DEHUM_CTRL_SVR_ATTR_RH_DISPLAY	Relative humidity display

5.8 Device temperature configuration cluster

```
#include "zcl/general/zcl.device.temp.h"
```

5.8.1 Device temperature configuration cluster description

Table 97. Device temperature configuration cluster PICS code description

PICS Code	Name	Supported	Notes
DTMP.S	Server	True	-
DTMP.C	Client	True	-
DTMP.S.A0000	`CurrentTemperature` server attribute	True	-
DTMP.S.A0001	`MinTempExperienced` server attribute	True	Optional
DTMP.S.A0002	`MaxTempExperienced` server attribute	True	Optional
DTMP.S.A0010	`DeviceTempAlarmMask` server attribute	True	Optional
DTMP.S.A0011	`LowTempThreshold` server attribute	True	Optional
DTMP.S.A0012	`HighTempThreshold` server attribute	True	Optional
DTMP.S.A0013	`LowTempDwellTripPoint` server attribute	True	Optional
DTMP.S.A0014	`HighTempDwellTripPoint` server attribute	True	Optional
DTMP.S.Afffd	`ClusterRevision` server attribute	True	-
DTMP.S.Afffe	`AttributeReportingStatus` server attribute	False	-
DTMP.C.Afffd	`ClusterRevision` client attribute	True	-
DTMP.C.Afffe	`AttributeReportingStatus` client attribute	False	-

5.8.2 Device temperature configuration cluster functions

5.8.2.1 *ZbZclDevTempClientAlloc*

```
struct ZbZclClusterT * ZbZclDevTempClientAlloc(struct ZigBeeT *zb, uint8_t endpoint);
```

Create a new instance of the Device Temperature Configuration Client cluster

Table 98. ZbZclDevTempClientAlloc parameters

Parameter	Description
`zb`	Zigbee® stack instance stack instance
`endpoint`	Endpoint on which to create cluster

Return: Cluster pointer, or NULL if there is an error.

5.8.2.2 *ZbZclDevTempServerAlloc*

```
struct ZbZclClusterT * ZbZclDevTempServerAlloc(struct ZigBeeT *zb, uint8_t endpoint);
```

Instantiate a new instance of the Device Temp client cluster.

Table 99. ZbZclDevTempServerAlloc parameters

Parameter	Description
`zb`	Zigbee® stack instance stack instance
`endpoint`	Endpoint on which to create cluster

Return: Cluster pointer, or NULL if there is an error.

5.8.3 Device temperature configuration cluster enumerations

5.8.3.1 *ZbZclDeviceTempAlarmCode*

Table 100. **Device Temperature Configuration Alarm Code**

Configuration code	Description
ZCL_DEV_TEMP_ALARM_CODE_LOW	Device Temperature too low
ZCL_DEV_TEMP_ALARM_CODE_HIGH	Device Temperature too high

5.8.3.2 *ZbZclDeviceTempAlarmMask*

Table 101. **Device Temperature Configuration Alarm Mask**

Configuration code	Description
ZCL_DEV_TEMP_ALARM_MASK_CLEAR	Alarm mask clear
ZCL_DEV_TEMP_ALARM_MASK_LOW	Alarm mask low
ZCL_DEV_TEMP_ALARM_MASK_HIGH	Alarm mask high

5.8.3.3 *ZbZclDeviceTempSvrAttrT*

Table 102. **Device Temperature Cluster Attribute IDs**

Cluster attribute ID	Description
ZCL_DEV_TEMP_CURRENT	Current temperature
ZCL_DEV_TEMP_MIN_TEMP	Min temp experienced (optional)
ZCL_DEV_TEMP_MAX_TEMP	Max temp experienced (optional)
ZCL_DEV_TEMP_OVER_TEMP_DWELL	Over temp total dwell (optional)
ZCL_DEV_TEMP_ALARM_MASK	Device temp alarm mask (optional)
ZCL_DEV_TEMP_LOW_THRESHOLD	Low temp threshold (optional)
ZCL_DEV_TEMP_HIGH_THRESHOLD	High temp threshold (optional)
ZCL_DEV_TEMP_LOW_DWELL_TRIP	Low temp dwell trip point (optional)
ZCL_DEV_TEMP_HIGH_DWELL_TRIP	High temp dwell trip point (optional)

5.9 Diagnostics cluster

```
#include "zcl/general/zcl.diagnostics.h"
```

5.9.1 Diagnostics cluster description

Table 103. **Diagnostics cluster PICS code description**

PICS Code	Name	Supported	Notes
DIAG.S	Server	True	-
DIAG.C	Client	True	-
DIAG.S.A0100	`MacRxBcast` server attribute	True	Optional
DIAG.S.A0101	`MacTxBcast` server attribute	True	Optional
DIAG.S.A0102	`MacRxUcast` server attribute	True	Optional

PICS Code	Name	Supported	Notes
DIAG.S.A0103	`MacTxUcast` server attribute	True	Optional
DIAG.S.A0104	`MacTxUcastRetry` server attribute	True	Optional
DIAG.S.A0105	`MacTxUcastFail` server attribute	True	Optional
DIAG.S.A0106	`APSRxBcast` server attribute	True	Optional
DIAG.S.A0107	`APSTxBcast` server attribute	True	Optional
DIAG.S.A0108	`APSRxUcast` server attribute	True	Optional
DIAG.S.A0109	`APSTxUcastSuccess` server attribute	True	Optional
DIAG.S.A010a	`APSTxUcastRetry` server attribute	True	Optional
DIAG.S.A010b	`APSTxUcastFail` server attribute	True	Optional
DIAG.S.A010c	`RouteDiscInitiated` server attribute	True	Optional
DIAG.S.A010d	`NeighborAdded` server attribute	True	Optional
DIAG.S.A010e	`NeighborRemoved` server attribute	True	Optional
DIAG.S.A010f	`NeighborStale` server attribute	True	Optional
DIAG.S.A0110	`JoinIndication` server attribute	True	Optional
DIAG.S.A0111	`ChildMoved` server attribute	True	Optional
DIAG.S.A0112	`NWKFCFailure` server attribute	True	Optional
DIAG.S.A0113	`APSFCFailure` server attribute	True	Optional
DIAG.S.A0114	`APSUnauthorizedKey` server attribute	True	Optional
DIAG.S.A0115	`NWKDecryptFailures` server attribute	True	Optional
DIAG.S.A0116	`APSDecryptFailures` server attribute	True	Optional
DIAG.S.A0117	`PacketBufferAllocateFailures` server attribute	True	Optional
DIAG.S.A0118	`RelayedUcast` server attribute	True	Optional
DIAG.S.A011a	`PacketValidatedropcount` server attribute	True	Optional
DIAG.S.A011c	`LastMessageLQI` server attribute	True	Optional
DIAG.S.A011d	`LastMessageRSSI` server attribute	True	Optional
DIAG.S.Afffd	`ClusterRevision` server attribute	True	-
DIAG.S.Afffe	`AttributeReportingStatus` server attribute	False	-
DIAG.C.Afffd	`ClusterRevision` client attribute	True	-
DIAG.C.Afffe	`AttributeReportingStatus` client attribute	False	-

5.9.2 Diagnostics cluster functions

5.9.2.1 *ZbZclDiagClientAlloc*

```
struct ZbZclClusterT * ZbZclDiagClientAlloc(struct ZigBeeT *zb, uint8_t endpoint);
```

Create a new instance of the Diagnostics Client cluster

Table 104. **ZbZclDiagClientAlloc parameters**

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster

Return: Cluster pointer, or NULL if there is an error.

5.9.2.2 *ZbZclDiagServerAlloc*

```
bool ZbZclDiagServerAlloc(struct ZigBeeT *zb, uint8_t endpoint, uint16_t profileId, enum
ZbStatusCodeT minSecurity);
```

Create a new instance of the Diagnostics Server cluster. Only one Diagnostics Server is allocated on the device

Table 105. **ZbZclDiagServerAlloc parameters**

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster
`profileId`	Profile ID setting, unless set to ZCL_PROFILE_WILDCARD
`minSecurity`	Minimum security level can be either: • ZB_APS_STATUS_UNSECURED • ZB_APS_STATUS_SECURED_NWK_KEY • ZB_APS_STATUS_SECURED_LINK_KEY

Return: True on success, false otherwise

5.9.3 Diagnostics cluster enumerations

5.9.3.1 *ZbZclDiagSvrAttrT*

Table 106. **Diagnostics server attribute IDs**

Attributes	Description
ZCL_DIAG_SVR_ATTR_RESETS	Number of resets (optional)
ZCL_DIAG_SVR_ATTR_PERSIST_WRITES	Persistent memory writes (optional)
ZCL_DIAG_SVR_ATTR_MAC_RX_BCAST	Mac Rx Bcast (optional)
ZCL_DIAG_SVR_ATTR_MAC_TX_BCAST	Mac Tx Bcast (optional)
ZCL_DIAG_SVR_ATTR_MAC_RX_UCAST	Mac Rx Ucast (optional)
ZCL_DIAG_SVR_ATTR_MAC_TX_UCAST	Mac Tx Ucast (optional)
ZCL_DIAG_SVR_ATTR_MAC_TX_UCAST_RETRY	Mac Tx Ucast retry (optional)
ZCL_DIAG_SVR_ATTR_MAC_TX_UCAST_FAIL	Mac Tx Ucast fail (optional)
ZCL_DIAG_SVR_ATTR_APS_RX_BCAST	APS Rx Bcast (optional)
ZCL_DIAG_SVR_ATTR_APS_TX_BCAST	APS Tx Bcast (optional)
ZCL_DIAG_SVR_ATTR_APS_RX_UCAST	APS Rx Ucast (optional)
ZCL_DIAG_SVR_ATTR_APS_TX_UCAST_SUCCESS	APS Tx Ucast success (optional)
ZCL_DIAG_SVR_ATTR_APS_TX_UCAST_RETRY	APS Tx Ucast retry (optional)
ZCL_DIAG_SVR_ATTR_APS_TX_UCAST_FAIL	APS Tx Ucast fail (optional)
ZCL_DIAG_SVR_ATTR_ROUTE_DISC_INIT	Route disc initiated (optional)
ZCL_DIAG_SVR_ATTR_NEIGHBOR_ADDED	Neighbor added (optional)
ZCL_DIAG_SVR_ATTR_NEIGHBOUR_REMOVED	Neighbor removed (optional)
ZCL_DIAG_SVR_ATTR_NEIGHBOUR_STALE	Neighbor stale (optional)
ZCL_DIAG_SVR_ATTR_JOIN_IND	Join indication (optional)

Attributes	Description
ZCL_DIAG_SVR_ATTR_CHILD_MOVED	Child moved (optional)
ZCL_DIAG_SVR_ATTR_NWK_FC_FAILURE	NWKFC failure (optional)
ZCL_DIAG_SVR_ATTR_APS_FC_FAILURE	APSFC failure (optional)
ZCL_DIAG_SVR_ATTR_APS_UNAUTH_KEY	APS unauthorized key (optional)
ZCL_DIAG_SVR_ATTR_NWK_DECRYPT_FAILS	NWK decrypt failures (optional)
ZCL_DIAG_SVR_ATTR_APS_DECRYPT_FAILS	APS decrypt failures (optional)
ZCL_DIAG_SVR_ATTR_PACKET_BUF_ALLOC_FAILS	Packet buffer allocate failures (optional)
ZCL_DIAG_SVR_ATTR_RELAYED_UCAST	Relayed Ucast (optional)
ZCL_DIAG_SVR_ATTR_PHY_MAC_QUEUE_LIM	Phyto MAC queue limit reached (optional)
ZCL_DIAG_SVR_ATTR_PACKET_VAL_DROP_COUNT	Packet validate drop count (optional)
ZCL_DIAG_SVR_ATTR_AVG_MAC_RETRY_PER_APS_MSG	Average MAC retry per APS message sent (optional)
ZCL_DIAG_SVR_ATTR_LAST_MSG_LQI	Last message LQI (optional)
ZCL_DIAG_SVR_ATTR_LAST_MSG_RSSI	Last message RSSI (optional)

5.10 Door lock cluster

```
#include "zcl/general/zcl.doorlock.h"
```

5.10.1 Door lock cluster description

Table 107. Door lock cluster PICS code description

PICS Code	Name	Supported	Notes
DRLK.S	Server	True	-
DRLK.C	Client	True	-
DRLK.S.A0000	LockState server attribute	True	-
DRLK.S.A0001	LockType server attribute	True	-
DRLK.S.A0002	ActuatorEnabled server attribute	True	-
DRLK.S.A0003	DoorState server attribute	True	Optional
DRLK.S.A0004	DoorOpenEvents server attribute	True	Optional
DRLK.S.A0005	DoorClosedEvents server attribute	True	Optional
DRLK.S.A0006	OpenPeriod server attribute	True	Optional
DRLK.S.A0010	NumberOfLogRecordsSupported server attribute	True	Optional
DRLK.S.A0011	NumberOfTotalUsersSupported server attribute	True	Optional
DRLK.S.A0012	NumberOfPINUsersSupported server attribute	True	Optional
DRLK.S.A0013	NumberOfRFIDUsersSupported server attribute	True	Optional
DRLK.S.A0014	NumberOfWeekDaySchedulesSupportedPerUser server attribute	True	Optional
DRLK.S.A0015	NumberOfYearDaySchedulesSupportedPerUser server attribute	True	Optional
DRLK.S.A0016	NumberOfHolidaySchedulesSupported server attribute	True	Optional
DRLK.S.A0017	MaxPINCodeLength server attribute	True	Optional
DRLK.S.A0018	MinPINCodeLength server attribute	True	Optional

PICS Code	Name	Supported	Notes
DRLK.S.A0019	`MaxRFIDCodeLength` server attribute	True	Optional
DRLK.S.A001a	`MinRFIDCodeLength` server attribute	True	Optional
DRLK.S.A0021	`Language` server attribute	True	Optional
DRLK.S.A0023	`AutoRelockTime` server attribute	True	Optional
DRLK.S.A0025	`OperatingMode` server attribute	True	Optional
DRLK.S.A0026	`SupportedOperatingModes` server attribute	True	Optional
DRLK.S.A0028	`EnableLocalProgramming` server attribute	True	Optional
DRLK.S.A0030	`WrongCodeEntryLimit` server attribute	True	Optional
DRLK.S.A0031	`UserCodeTemporaryDisableTime` server attribute	True	Optional
DRLK.S.A0033	`RequirePINforRFOperation` server attribute	True	Optional
DRLK.S.A0034	`SecurityLevel` server attribute	True	Optional
DRLK.S.A0040	`AlarmMask` server attribute	True	Optional
DRLK.S.Afffd	`ClusterRevision` server attribute	True	-
DRLK.S.Afffe	`AttributeReportingStatus` server attribute	False	-
DRLK.S.C00.Rsp	Lock Door server command	True	-
DRLK.S.C01.Rsp	Unlock Door server command	True	-
DRLK.S.C02.Rsp	Toggle server command	True	-
DRLK.S.C03.Rsp	Unlock with Timeout server command	False	-
DRLK.S.C04.Rsp	Get Log Record server command	False	-
DRLK.S.C05.Rsp	Set PIN Code server command	False	-
DRLK.S.C06.Rsp	Get PIN Code server command	False	-
DRLK.S.C07.Rsp	Clear PIN Code server command	False	-
DRLK.S.C08.Rsp	Clear All PIN Codes server command	False	-
DRLK.S.C09.Rsp	Set User Status server command	False	-
DRLK.S.C0a.Rsp	Get User Status server command	False	-
DRLK.S.C0b.Rsp	Set Weekday Schedule server command	False	-
DRLK.S.C0c.Rsp	Get Weekday Schedule server command	False	-
DRLK.S.C0d.Rsp	Clear Weekday Schedule server command	False	-
DRLK.S.C0e.Rsp	Set Year Day Schedule server command	False	-
DRLK.S.C0f.Rsp	Get Year Day Schedule server command	False	-
DRLK.S.C10.Rsp	Clear Year Day Schedule server command	False	-
DRLK.S.C11.Rsp	Set Holiday Schedule server command	False	-
DRLK.S.C12.Rsp	Get Holiday Schedule server command	False	-
DRLK.S.C13.Rsp	Clear Holiday Schedule server command	False	-
DRLK.S.C14.Rsp	Set User Type server command	False	-
DRLK.S.C15.Rsp	Get User Type server command	False	-
DRLK.S.C16.Rsp	Set RFID Code server command	False	-
DRLK.S.C17.Rsp	Get RFID Code server command	False	-
DRLK.S.C18.Rsp	Clear RFID Code server command	False	-
DRLK.S.C19.Rsp	Clear All RFID Codes server command	False	-

PICS Code	Name	Supported	Notes
DRLK.S.C00.Tx	Lock Door Response server command	True	-
DRLK.S.C01.Tx	Unlock Door Response server command	True	-
DRLK.S.C02.Tx	Toggle Response server command	True	-
DRLK.S.C03.Tx	Unlock with Timeout Response server command	False	-
DRLK.S.C04.Tx	Get Log Record Response server command	False	-
DRLK.S.C05.Tx	Set PIN Code Response server command	False	-
DRLK.S.C06.Tx	Get PIN Code Response server command	False	-
DRLK.S.C07.Tx	Clear PIN Code Response server command	False	-
DRLK.S.C08.Tx	Clear All PIN Codes Response server command	False	-
DRLK.S.C09.Tx	Set User Status Response server command	False	-
DRLK.S.C0a.Tx	Get User Status Response server command	False	-
DRLK.S.C0b.Tx	Set Weekday Schedule Response server command	False	-
DRLK.S.C0c.Tx	Get Weekday Schedule Response server command	False	-
DRLK.S.C0d.Tx	Clear Weekday Schedule Response server command	False	-
DRLK.S.C0e.Tx	Set Year Day Schedule Response server command	False	-
DRLK.S.C0f.Tx	Get Year Day Schedule Response server command	False	-
DRLK.S.C10.Tx	Clear Year Day Schedule Response server command	False	-
DRLK.S.C11.Tx	Set Holiday Schedule Response server command	False	-
DRLK.S.C12.Tx	Get Holiday Schedule Response server command	False	-
DRLK.S.C13.Tx	Clear Holiday Schedule Response server command	False	-
DRLK.S.C14.Tx	Set User Type Response server command	False	-
DRLK.S.C15.Tx	Get User Type Response server command	False	-
DRLK.S.C16.Tx	Set RFID Code Response server command	False	-
DRLK.S.C17.Tx	Get RFID Code Response server command	False	-
DRLK.S.C18.Tx	Clear RFID Code Response server command	False	-
DRLK.S.C19.Tx	Clear All RFID Codes Response server command	False	-
DRLK.S.C20.Tx	Operating Event Notification server command	False	-
DRLK.S.C21.Tx	Programming Event Notification server command	False	-
DRLK.C.Afffd	`ClusterRevision` client attribute	True	-
DRLK.C.Afffe	`AttributeReportingStatus` client attribute	False	-
DRLK.C.C00.Rsp	Lock Door Response client command	True	-
DRLK.C.C01.Rsp	Unlock Door Response client command	True	-
DRLK.C.C02.Rsp	Toggle Response client command	True	-
DRLK.C.C03.Rsp	Unlock with Timeout Response client command	False	-
DRLK.C.C04.Rsp	Get Log Record Response client command	False	-
DRLK.C.C05.Rsp	Set PIN Code Response client command	False	-
DRLK.C.C06.Rsp	Get PIN Code Response client command	False	-
DRLK.C.C07.Rsp	Clear PIN Code Response client command	False	-
DRLK.C.C08.Rsp	Clear All PIN Codes Response client command	False	-
DRLK.C.C09.Rsp	Set User Status Response client command	False	-
DRLK.C.C0a.Rsp	Get User Status Response client command	False	-

PICS Code	Name	Supported	Notes
DRLK.C.C0b.Rsp	Set Weekday Schedule Response client command	False	-
DRLK.C.C0c.Rsp	Get Weekday Schedule Response client command	False	-
DRLK.C.C0d.Rsp	Clear Weekday Schedule Response client command	False	-
DRLK.C.C0e.Rsp	Set Year Day Schedule Response client command	False	-
DRLK.C.C0f.Rsp	Get Year Day Schedule Response client command	False	-
DRLK.C.C10.Rsp	Clear Year Day Schedule Response client command	False	-
DRLK.C.C11.Rsp	Set Holiday Schedule Response client command	False	-
DRLK.C.C12.Rsp	Get Holiday Schedule Response client command	False	-
DRLK.C.C13.Rsp	Clear Holiday Schedule Response client command	False	-
DRLK.C.C14.Rsp	Set User Type Response client command	False	-
DRLK.C.C15.Rsp	Get User Type Response client command	False	-
DRLK.C.C16.Rsp	Set RFID Code Response client command	False	-
DRLK.C.C17.Rsp	Get RFID Code Response client command	False	-
DRLK.C.C18.Rsp	Clear RFID Code Response client command	False	-
DRLK.C.C19.Rsp	Clear All RFID Codes Response client command	False	-
DRLK.C.C20.Rsp	Operating Event Notification client command	False	-
DRLK.C.C21.Rsp	Programming Event Notification client command	False	-
DRLK.C.C00.Tx	Lock Door client command	True	-
DRLK.C.C01.Tx	Unlock Door client command	True	-
DRLK.C.C02.Tx	Toggle client command	True	-
DRLK.C.C03.Tx	Unlock with Timeout client command	False	-
DRLK.C.C04.Tx	Get Log Record client command	False	-
DRLK.C.C05.Tx	Set PIN Code client command	False	-
DRLK.C.C06.Tx	Get PIN Code client command	False	-
DRLK.C.C07.Tx	Clear PIN Code client command	False	-
DRLK.C.C08.Tx	Clear All PIN Codes client command	False	-
DRLK.C.C09.Tx	Set User Status client command	False	-
DRLK.C.C0a.Tx	Get User Status client command	False	-
DRLK.C.C0b.Tx	Set Weekday Schedule client command	False	-
DRLK.C.C0c.Tx	Get Weekday Schedule client command	False	-
DRLK.C.C0d.Tx	Clear Weekday Schedule client command	False	-
DRLK.C.C0e.Tx	Set Year Day Schedule client command	False	-
DRLK.C.C0f.Tx	Get Year Day Schedule client command	False	-
DRLK.C.C10.Tx	Clear Year Day Schedule client command	False	-
DRLK.C.C11.Tx	Set Holiday Schedule client command	False	-
DRLK.C.C12.Tx	Get Holiday Schedule client command	False	-
DRLK.C.C13.Tx	Clear Holiday Schedule client command	False	-
DRLK.C.C14.Tx	Set User Type client command	False	-
DRLK.C.C15.Tx	Get User Type client command	False	-
DRLK.C.C16.Tx	Get RFID Code client command	False	-
DRLK.C.C17.Tx	Get RFID Code client command	False	-

PICS Code	Name	Supported	Notes
DRLK.C.C18.Tx	Clear RFID Code client command	False	-
DRLK.C.C19.Tx	Clear All RFID Codes client command	False	-

5.10.2 Door lock cluster functions

5.10.2.1 *ZbZclDoorLockClientAlloc*

```
struct ZbZclClusterT * ZbZclDoorLockClientAlloc(struct ZigBeeT *zb, uint8_t endpoint);
```

Create a new instance of the Door Lock Client cluster

Table 108. ZbZclDoorLockClientAlloc parameters

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster

Return: Cluster pointer, or NULL if there is an error.

5.10.2.2 *ZbZclDoorLockClientClrAllPinReq*

```
enum ZclStatusCodeT ZbZclDoorLockClientClrAllPinReq(struct ZbZclClusterT *cluster, const
struct ZbApsAddrT *dst, void (*callback)(struct ZbZclCommandRspT *rsp, void*arg), void *arg);
```

Send a Clear All PIN Codes request command

Table 109. ZbZclDoorLockClientClrAllPinReq parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.10.2.3 *ZbZclDoorLockClientClrAllRfidReq*

```
enum ZclStatusCodeT ZbZclDoorLockClientClrAllRfidReq(struct ZbZclClusterT *cluster, const
struct ZbApsAddrT *dst, void (*callback)(struct ZbZclCommandRspT *rsp, void*arg), void *arg);
```

Send a Clear All RFID Codes request command

Table 110. ZbZclDoorLockClientClrAllRfidReq parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.10.2.4 ZbZclDoorLockClientClrHDScheduleReq

```
enum ZclStatusCodeT ZbZclDoorLockClientClrHDScheduleReq(struct ZbZclClusterT*cluster, const
struct ZbApsAddrT *dst, struct ZbZclDoorLockClrHDScheduleReqT *req, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Clear Holiday Schedule request command

Table 111. ZbZclDoorLockClientClrHDScheduleReq parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`req`	Get Year Day Schedule request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.10.2.5 ZbZclDoorLockClientClrPinReq

```
enum ZclStatusCodeT ZbZclDoorLockClientClrPinReq(struct ZbZclClusterT *cluster, const
struct ZbApsAddrT *dst, struct ZbZclDoorLockClrPinReqT *req, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Clear PIN Code request command

Table 112. ZbZclDoorLockClientClrPinReq parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`req`	Clear PIN Code request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.10.2.6 ZbZclDoorLockClientClrRfidReq

```
enum ZclStatusCodeT ZbZclDoorLockClientClrRfidReq(struct ZbZclClusterT *cluster, const
struct ZbApsAddrT *dst, struct ZbZclDoorLockClrRfidReqT *req, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Clear RFID Code request command

Table 113. ZbZclDoorLockClientClrRfidReq parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`req`	Clear RFID Code request structure
`callback`	Callback function that is invoked when the response is received

Parameter	Description
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.10.2.7 *ZbZclDoorLockClientClrWDScheduleReq*

```
enum ZclStatusCodeT ZbZclDoorLockClientClrWDScheduleReq(struct ZbZclClusterT*cluster, const
struct ZbApsAddrT *dst, struct ZbZclDoorLockClrWDScheduleReqT *req, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Clear Weekday Schedule request command

Table 114. **ZbZclDoorLockClientClrWDScheduleReq parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`req`	Clear Weekday Schedule request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.10.2.8 *ZbZclDoorLockClientClrYDScheduleReq*

```
enum ZclStatusCodeT ZbZclDoorLockClientClrYDScheduleReq(struct ZbZclClusterT*cluster, const
struct ZbApsAddrT *dst, struct ZbZclDoorLockClrYDScheduleReqT *req, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Clear Year Day Schedule request command

Table 115. **ZbZclDoorLockClientClrYDScheduleReq parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`req`	Clear Year Day Schedule request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.10.2.9 *ZbZclDoorLockClientGetLogReq*

```
enum ZclStatusCodeT ZbZclDoorLockClientGetLogReq(struct ZbZclClusterT *cluster, const
struct ZbApsAddrT *dst, struct ZbZclDoorLockGetLogReqT *req, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Get Log Record request command

Table 116. ZbZclDoorLockClientGetLogReq parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`req`	Get Log Record request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.10.2.10 *ZbZclDoorLockClientGetPinReq*

```
enum ZclStatusCodeT ZbZclDoorLockClientGetPinReq(struct ZbZclClusterT *cluster, const
struct ZbApsAddrT *dst, struct ZbZclDoorLockGetPinReqT *req, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Get PIN Code request command

Table 117. ZbZclDoorLockClientGetPinReq Parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`req`	Get PIN Code request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.10.2.11 *ZbZclDoorLockClientGetRfidReq*

```
enum ZclStatusCodeT ZbZclDoorLockClientGetRfidReq(struct ZbZclClusterT *cluster, const
struct ZbApsAddrT *dst, struct ZbZclDoorLockGetRfidReqT *req, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Get RFID Code request command

Table 118. ZbZclDoorLockClientGetRfidReq parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`req`	Get RFID Code request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.10.2.12 *ZbZclDoorLockClientGetUserStatusReq*

```
enum ZclStatusCodeT ZbZclDoorLockClientGetUserStatusReq(struct ZbZclClusterT*cluster, const
struct ZbApsAddrT *dst, struct ZbZclDoorLockGetUserStatusReqT *req, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Get User Status request command

Table 119. ZbZclDoorLockClientGetUserStatusReq parameters

Parameter	Description
cluster	Cluster instance from which to send this command
dst	Requested destination address
req	Get User Status request structure
callback	Callback function that is invoked when the response is received
arg	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.10.2.13 *ZbZclDoorLockClientGetHDScheduleReq*

```
enum ZclStatusCodeT ZbZclDoorLockClientGetHDScheduleReq(struct ZbZclClusterT*cluster, const
struct ZbApsAddrT *dst, struct ZbZclDoorLockGetHDScheduleReqT *req, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Get Holiday Schedule request command

Table 120. ZbZclDoorLockClientGetHDScheduleReq parameters

Parameter	Description
cluster	Cluster instance from which to send this command
dst	Requested destination address
req	Get Holiday Schedule request structure
callback	Callback function that is invoked when the response is received
arg	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.10.2.14 *ZbZclDoorLockClientGetUserTypeReq*

```
enum ZclStatusCodeT ZbZclDoorLockClientGetUserTypeReq(struct ZbZclClusterT*cluster, const
struct ZbApsAddrT *dst, struct ZbZclDoorLockGetUserTypeReqT *req, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Get User Type request command

Table 121. ZbZclDoorLockClientGetUserTypeReq parameters

Parameter	Description
cluster	Cluster instance from which to send this command
dst	Requested destination address
req	Get User Type request structure
callback	Callback function that is invoked when the response is received

Parameter	Description
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.10.2.15 *ZbZclDoorLockClientGetWDScheduleReq*

```
enum ZclStatusCodeT ZbZclDoorLockClientGetWDScheduleReq(struct ZbZclClusterT*cluster, const
struct ZbApsAddrT *dst, struct ZbZclDoorLockGetWDScheduleReqT *req, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Get Weekday Schedule request command

Table 122. **ZbZclDoorLockClientGetWDScheduleReq parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`req`	Get Weekday Schedule request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.10.2.16 *ZbZclDoorLockClientGetYDScheduleReq*

```
enum ZclStatusCodeT ZbZclDoorLockClientGetYDScheduleReq(struct ZbZclClusterT*cluster, const
struct ZbApsAddrT *dst, struct ZbZclDoorLockGetYDScheduleReqT *req, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Get Year Day Schedule request command

Table 123. **ZbZclDoorLockClientGetYDScheduleReq parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`req`	Get Year Day Schedule request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.10.2.17 *ZbZclDoorLockClientLockReq*

```
enum ZclStatusCodeT ZbZclDoorLockClientLockReq(struct ZbZclClusterT *cluster, const
struct ZbApsAddrT *dst, struct ZbZclDoorLockLockDoorReqT *req, void (*callback)(struct
ZbZclCommandRspT *zcl_rsp, void *arg), void *arg);
```

Send a Lock Door request command

Table 124. **ZbZclDoorLockClientLockReq Parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`req`	Lock door request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.10.2.18 *ZbZclDoorLockClientSetHDScheduleReq*

```
enum ZclStatusCodeT ZbZclDoorLockClientSetHDScheduleReq(struct ZbZclClusterT*cluster, const
struct ZbApsAddrT *dst, struct ZbZclDoorLockSetHDScheduleReqT *req, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Set Holiday Schedule request command

Table 125. **ZbZclDoorLockClientSetHDScheduleReq parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`req`	Set Holiday Schedule request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.10.2.19 *ZbZclDoorLockClientSetPinReq*

```
enum ZclStatusCodeT ZbZclDoorLockClientSetPinReq(struct ZbZclClusterT *cluster, const
struct ZbApsAddrT *dst, struct ZbZclDoorLockSetPinReqT *req, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Set PIN Code request command

Table 126. **ZbZclDoorLockClientSetPinReq parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`req`	Set PIN Code request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.10.2.20 *ZbZclDoorLockClientSetRfidReq*

```
enum ZclStatusCodeT ZbZclDoorLockClientSetRfidReq(struct ZbZclClusterT *cluster, const
struct ZbApsAddrT *dst, struct ZbZclDoorLockSetRfidReqT *req, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Set RFID Code request command

Table 127. ZbZclDoorLockClientSetRfidReq Parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`req`	Set RFID Code request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.10.2.21 *ZbZclDoorLockClientSetUserStatusReq*

```
enum ZclStatusCodeT ZbZclDoorLockClientSetUserStatusReq(struct ZbZclClusterT*cluster, const
struct ZbApsAddrT *dst, struct ZbZclDoorLockSetUserStatusReqT *req, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Set User Status request command

Table 128. ZbZclDoorLockClientSetUserStatusReq parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`req`	Set User Status request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.10.2.22 *ZbZclDoorLockClientSetUserTypeReq*

```
enum ZclStatusCodeT ZbZclDoorLockClientSetUserTypeReq(struct ZbZclClusterT*cluster, const
struct ZbApsAddrT *dst, struct ZbZclDoorLockSetUserTypeReqT *req, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Set User Type request command

Table 129. ZbZclDoorLockClientSetUserTypeReq parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`req`	Set User Type request structure
`callback`	Callback function that is invoked when the response is received

Parameter	Description
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.10.2.23 *ZbZclDoorLockClientSetWDScheduleReq*

```
enum ZclStatusCodeT ZbZclDoorLockClientSetWDScheduleReq(struct ZbZclClusterT*cluster, const
struct ZbApsAddrT *dst, struct ZbZclDoorLockSetWDScheduleReqT *req, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Set Weekday Schedule request command

Table 130. **ZbZclDoorLockClientSetWDScheduleReq parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`req`	Set Weekday Schedule request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.10.2.24 *ZbZclDoorLockClientSetYDScheduleReq*

```
enum ZclStatusCodeT ZbZclDoorLockClientSetYDScheduleReq(struct ZbZclClusterT*cluster, const
struct ZbApsAddrT *dst, struct ZbZclDoorLockSetYDScheduleReqT *req, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Set Year Day Schedule request command

Table 131. **ZbZclDoorLockClientSetYDScheduleReq parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`req`	Set Year Day Schedule request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.10.2.25 *ZbZclDoorLockClientToggleReq*

```
enum ZclStatusCodeT ZbZclDoorLockClientToggleReq(struct ZbZclClusterT *cluster, const
struct ZbApsAddrT *dst, struct ZbZclDoorLockToggleReqT *req, void (*callback)(struct
ZbZclCommandRspT *zcl_rsp, void *arg), void *arg);
```

Send Toggle request command

Table 132. **ZbZclDoorLockClientToggleReq parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`req`	Toggle request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.10.2.26 *ZbZclDoorLockClientUnlockReq*

```
enum ZclStatusCodeT ZbZclDoorLockClientUnlockReq(struct ZbZclClusterT *cluster, const
struct ZbApsAddrT *dst, struct ZbZclDoorLockUnlockDoorReqT *req, void (*callback)(struct
ZbZclCommandRspT *zcl_rsp, void *arg), void *arg);
```

Send an Unlock Door request command

Table 133. **ZbZclDoorLockClientUnlockReq parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`req`	Unlock Door request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.10.2.27 *ZbZclDoorLockClientUnlockTimeoutReq*

```
enum ZclStatusCodeT ZbZclDoorLockClientUnlockTimeoutReq(struct ZbZclClusterT*cluster, const
struct ZbApsAddrT *dst, struct ZbZclDoorLockUnlockTimeoutReqT *req, void (*callback)(struct
ZbZclCommandRspT *zcl_rsp, void *arg), void *arg);
```

Send an Unlock with Timeout request command

Table 134. **ZbZclDoorLockClientUnlockTimeoutReq parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`req`	Unlock with Timeout request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.10.2.28 *ZbZclDoorLockServerAlloc*

```
struct ZbZclClusterT * ZbZclDoorLockServerAlloc(struct ZigBeeT *zb, uint8_t endpoint, struct
ZbZclDoorLockServerCallbacksT *callbacks, void *arg);
```

Create a new instance of the Door Lock Server cluster

Table 135. ZbZclDoorLockServerAlloc parameters

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: Cluster pointer, or NULL if there is an error.

5.10.2.29 *ZbZclDoorLockServerSendClrAllPinRsp*

```
enum ZclStatusCodeT ZbZclDoorLockServerSendClrAllPinRsp(struct ZbZclClusterT*cluster, struct
ZbZclAddrInfoT *dst, struct ZbZclDoorLockClrAllPinRspT *rsp, void (*callback)(struct
ZbApsdeDataConfT *conf, void *arg), void *arg);
```

Send a Clear All PIN Codes response command

Table 136. ZbZclDoorLockServerSendClrAllPinRsp parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`rsp`	Clear All PIN Codes response structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.10.2.30 *ZbZclDoorLockServerSendClrAllRfidRsp*

```
enum ZclStatusCodeT ZbZclDoorLockServerSendClrAllRfidRsp(struct ZbZclClusterT*cluster,
struct ZbZclAddrInfoT *dst, struct ZbZclDoorLockClrAllRfidRspT *rsp, void (*callback)(struct
ZbApsdeDataConfT *conf, void *arg), void *arg);
```

Send a Clear All RFID Codes response command

Table 137. ZbZclDoorLockServerSendClrAllRfidRsp parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`rsp`	Clear All RFID Codes response structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.10.2.31 *ZbZclDoorLockServerSendClrHDScheduleRsp*

```
enum ZclStatusCodeT ZbZclDoorLockServerSendClrHDScheduleRsp(struct ZbZclClusterT *cluster,
struct ZbZclAddrInfoT *dst, struct ZbZclDoorLockClrHDScheduleRspT *rsp, void (*callback)
(struct ZbApsdeDataConfT *conf, void *arg), void *arg);
```

Send a Clear Holiday Schedule response command

Table 138. ZbZclDoorLockServerSendClrHDScheduleRsp parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`rsp`	Clear Holiday Schedule response structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.10.2.32 *ZbZclDoorLockServerSendClrPinRsp*

```
enum ZclStatusCodeT ZbZclDoorLockServerSendClrPinRsp(struct ZbZclClusterT *cluster, struct
ZbZclAddrInfoT *dst, struct ZbZclDoorLockClrPinRspT *rsp, void (*callback)(struct
ZbApsdeDataConfT *conf, void *arg), void *arg);
```

Send a Clear PIN Code response command

Table 139. ZbZclDoorLockServerSendClrPinRsp parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`rsp`	Clear PIN code response structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.10.2.33 *ZbZclDoorLockServerSendClrRfidRsp*

```
enum ZclStatusCodeT ZbZclDoorLockServerSendClrRfidRsp(struct ZbZclClusterT*cluster, struct
ZbZclAddrInfoT *dst, struct ZbZclDoorLockClrRfidRspT *rsp, void (*callback)(struct
ZbApsdeDataConfT *conf, void *arg), void *arg);
```

Send a Clear RFID Code response command

Table 140. ZbZclDoorLockServerSendClrRfidRsp parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`rsp`	Clear RFID code response structure
`callback`	Callback function that is invoked when the response is received

Parameter	Description
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.10.2.34 ZbZclDoorLockServerSendClrWDScheduleRsp

```
enum ZclStatusCodeT ZbZclDoorLockServerSendClrWDScheduleRsp(struct ZbZclClusterT*cluster,
struct ZbZclAddrInfoT *dst, struct ZbZclDoorLockClrWDScheduleRspT *rsp, void (*callback)
(struct ZbApsdeDataConfT *conf, void *arg), void *arg);
```

Send a Clear Weekday Schedule response command

Table 141. ZbZclDoorLockServerSendClrWDScheduleRsp parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`rsp`	Clear Weekday Schedule response structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.10.2.35 ZbZclDoorLockServerSendClrYDScheduleRsp

```
enum ZclStatusCodeT ZbZclDoorLockServerSendClrYDScheduleRsp(struct ZbZclClusterT*cluster,
struct ZbZclAddrInfoT *dst, struct ZbZclDoorLockClrYDScheduleRspT *rsp, void (*callback)
(struct ZbApsdeDataConfT *conf, void *arg), void *arg);
```

Send a Clear Year Day Schedule response command

Table 142. ZbZclDoorLockServerSendClrYDScheduleRsp parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`rsp`	Clear Year Day Schedule response structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.10.2.36 ZbZclDoorLockServerSendGetHDScheduleRsp

```
enum ZclStatusCodeT ZbZclDoorLockServerSendGetHDScheduleRsp(struct ZbZclClusterT*cluster,
struct ZbZclAddrInfoT *dst, struct ZbZclDoorLockGetHDScheduleRspT *rsp, void (*callback)
(struct ZbApsdeDataConfT *conf, void *arg), void *arg);
```

Send a Get Holiday Schedule response command

Table 143. ZbZclDoorLockServerSendGetHDScheduleRsp parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Destination address for response, including sequence number and tx options
`rsp`	Get Holiday Schedule response structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.10.2.37 *ZbZclDoorLockServerSendGetLogRsp*

```
enum ZclStatusCodeT ZbZclDoorLockServerSendGetLogRsp(struct ZbZclClusterT *cluster, struct
ZbZclAddrInfoT *dst, struct ZbZclDoorLockGetLogRspT *rsp, void (*callback)(struct
ZbApsdeDataConfT *conf, void *arg), void *arg);
```

Send a Get Log Record response command

Table 144. ZbZclDoorLockServerSendGetLogRsp parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`rsp`	Get Log Record response structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.10.2.38 *ZbZclDoorLockServerSendGetPinRsp*

```
enum ZclStatusCodeT ZbZclDoorLockServerSendGetPinRsp(struct ZbZclClusterT *cluster, struct
ZbZclAddrInfoT *dst, struct ZbZclDoorLockGetPinRspT *rsp, void (*callback)(struct
ZbApsdeDataConfT *conf, void *arg), void *arg);
```

Send a Get PIN Code response command

Table 145. ZbZclDoorLockServerSendGetPinRsp parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`rsp`	Get PIN Code response structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.10.2.39 *ZbZclDoorLockServerSendGetRfidRsp*

```
enum ZclStatusCodeT ZbZclDoorLockServerSendGetRfidRsp(struct ZbZclClusterT*cluster, struct
ZbZclAddrInfoT *dst, struct ZbZclDoorLockGetRfidRspT *rsp, void (*callback)(struct
ZbApsdeDataConfT *conf, void *arg), void *arg);
```

Send a Get RFID Code response command

Table 146. ZbZclDoorLockServerSendGetRfidRsp Parameters

Parameter	Description
cluster	Cluster instance from which to send this command
dst	Requested destination address
rsp	Get RFID Code response structure
callback	Callback function that is invoked when the response is received
arg	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.10.2.40 *ZbZclDoorLockServerSendGetUserStatusRsp*

```
enum ZclStatusCodeT ZbZclDoorLockServerSendGetUserStatusRsp(struct ZbZclClusterT *cluster,
struct ZbZclAddrInfoT *dst, struct ZbZclDoorLockGetUserStatusRspT *rsp, void (*callback)
(struct ZbApsdeDataConfT *conf, void *arg), void *arg);
```

Send a Get User Status response command

Table 147. ZbZclDoorLockServerSendGetUserStatusRsp parameters

Parameter	Description
cluster	Cluster instance from which to send this command
dst	Requested destination address
rsp	Get User Status response structure
callback	Callback function that is invoked when the response is received
arg	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.10.2.41 *ZbZclDoorLockServerSendGetUserTypeRsp*

```
enum ZclStatusCodeT ZbZclDoorLockServerSendGetUserTypeRsp(struct ZbZclClusterT*cluster,
struct ZbZclAddrInfoT *dst, struct ZbZclDoorLockGetUserTypeRspT *rsp, void (*callback)
(struct ZbApsdeDataConfT *conf, void *arg), void *arg);
```

Send a Get User Type response command

Table 148. ZbZclDoorLockServerSendGetUserTypeRsp parameters

Parameter	Description
cluster	Cluster instance from which to send this command
dst	Requested destination address
rsp	Get User Type response structure
callback	Callback function that is invoked when the response is received

Parameter	Description
arg	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.10.2.42 *ZbZclDoorLockServerSendGetWDScheduleRsp*

```
enum ZclStatusCodeT ZbZclDoorLockServerSendGetWDScheduleRsp(struct ZbZclClusterT*cluster,
struct ZbZclAddrInfoT *dst, struct ZbZclDoorLockGetWDScheduleRspT *rsp, void (*callback)
(struct ZbApsdeDataConfT *conf, void *arg), void *arg);
```

Send a Get Weekday Schedule response command

Table 149. **ZbZclDoorLockServerSendGetWDScheduleRsp parameters**

Parameter	Description
cluster	Cluster instance from which to send this command
dst	Requested destination address
rsp	Get Weekday Schedule response structure
callback	Callback function that is invoked when the response is received
arg	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.10.2.43 *ZbZclDoorLockServerSendGetYDScheduleRsp*

```
enum ZclStatusCodeT ZbZclDoorLockServerSendGetYDScheduleRsp(struct ZbZclClusterT*cluster,
struct ZbZclAddrInfoT *dst, struct ZbZclDoorLockGetYDScheduleRspT *rsp, void (*callback)
(struct ZbApsdeDataConfT *conf, void *arg), void *arg);
```

Send a Get Year Day Schedule response command

Table 150. **ZbZclDoorLockServerSendGetYDScheduleRsp parameters**

Parameter	Description
cluster	Cluster instance from which to send this command
dst	Requested destination address
rsp	Get Year Day Schedule response structure
callback	Callback function that is invoked when the response is received
arg	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.10.2.44 *ZbZclDoorLockServerSendLockRsp*

```
enum ZclStatusCodeT ZbZclDoorLockServerSendLockRsp(struct ZbZclClusterT *cluster, struct
ZbZclAddrInfoT *dst, struct ZbZclDoorLockLockDoorRspT *rsp, void (*callback)(struct
ZbApsdeDataConfT *conf, void *arg), void *arg);
```

Send a Lock Door response command

Table 151. **ZbZclDoorLockServerSendLockRsp parameters**

Parameter	Description
cluster	Cluster instance from which to send this command
dst	Requested destination address
rsp	Lock Door response structure
callback	Callback function that is invoked when the response is received
arg	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.10.2.45 *ZbZclDoorLockServerSendSetHDScheduleRsp*

```
enum ZclStatusCodeT ZbZclDoorLockServerSendSetHDScheduleRsp(struct ZbZclClusterT *cluster,
struct ZbZclAddrInfoT *dst, struct ZbZclDoorLockSetHDScheduleRspT *rsp, void (*callback)
(struct ZbApsdeDataConfT *conf, void *arg), void *arg);
```

Send a Set Holiday Schedule response command

Table 152. **ZbZclDoorLockServerSendSetHDScheduleRsp parameters**

Parameter	Description
cluster	Cluster instance from which to send this command
dst	Requested destination address
rsp	Set Holiday Schedule response structure
callback	Callback function that is invoked when the response is received
arg	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.10.2.46 *ZbZclDoorLockServerSendSetPinRsp*

```
enum ZclStatusCodeT ZbZclDoorLockServerSendSetPinRsp(struct ZbZclClusterT *cluster, struct
ZbZclAddrInfoT *dst, struct ZbZclDoorLockSetPinRspT *rsp, void (*callback)(struct
ZbApsdeDataConfT *conf, void *arg), void *arg);
```

Send a Set PIN Code response command

Table 153. **ZbZclDoorLockServerSendSetPinRsp parameters**

Parameter	Description
cluster	Cluster instance from which to send this command
dst	Requested destination address
rsp	Set PIN Code response structure
callback	Callback function that is invoked when the response is received
arg	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.10.2.47 *ZbZclDoorLockServerSendSetRfidRsp*

```
enum ZclStatusCodeT ZbZclDoorLockServerSendSetRfidRsp(struct ZbZclClusterT*cluster, struct
ZbZclAddrInfoT *dst, struct ZbZclDoorLockSetRfidRspT *rsp, void (*callback)(struct
ZbApsdeDataConfT *conf, void *arg), void *arg);
```

Send a Set RFID Code response command

Table 154. ZbZclDoorLockServerSendSetRfidRsp parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`rsp`	Set RFID Code response structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.10.2.48 *ZbZclDoorLockServerSendSetUserStatusRsp*

```
enum ZclStatusCodeT ZbZclDoorLockServerSendSetUserStatusRsp(struct ZbZclClusterT*cluster,
struct ZbZclAddrInfoT *dst, struct ZbZclDoorLockSetUserStatusRspT *rsp, void (*callback)
(struct ZbApsdeDataConfT *conf, void *arg), void *arg);
```

Send a Set User Status response command

Table 155. ZbZclDoorLockServerSendSetUserStatusRsp parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`rsp`	Set User Status response structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.10.2.49 *ZbZclDoorLockServerSendSetUserTypeRsp*

```
enum ZclStatusCodeT ZbZclDoorLockServerSendSetUserTypeRsp(struct ZbZclClusterT *cluster,
struct ZbZclAddrInfoT *dst, struct ZbZclDoorLockSetUserTypeRspT *rsp, void (*callback)
(struct ZbApsdeDataConfT *conf, void *arg), void *arg);
```

Send a Set User Type response command

Table 156. ZbZclDoorLockServerSendSetUserTypeRsp parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`rsp`	Set User Type response structure
`callback`	Callback function that is invoked when the response is received

Parameter	Description
arg	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.10.2.50 *ZbZclDoorLockServerSendSetWDScheduleRsp*

```
enum ZclStatusCodeT ZbZclDoorLockServerSendSetWDScheduleRsp(struct ZbZclClusterT*cluster,
struct ZbZclAddrInfoT *dst, struct ZbZclDoorLockSetWDScheduleRspT *rsp, void (*callback)
(struct ZbApsdeDataConfT *conf, void *arg), void *arg);
```

Send a Set Weekday Schedule response command

Table 157. ZbZclDoorLockServerSendSetWDScheduleRsp parameters

Parameter	Description
cluster	Cluster instance from which to send this command
dst	Requested destination address
rsp	Set Weekday Schedule response structure
callback	Callback function that is invoked when the response is received
arg	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.10.2.51 *ZbZclDoorLockServerSendSetYDScheduleRsp*

```
enum ZclStatusCodeT ZbZclDoorLockServerSendSetYDScheduleRsp(struct ZbZclClusterT*cluster,
struct ZbZclAddrInfoT *dst, struct ZbZclDoorLockSetYDScheduleRspT *rsp, void (*callback)
(struct ZbApsdeDataConfT *conf, void *arg), void *arg);
```

Send a Set Year Day Schedule response command

Table 158. ZbZclDoorLockServerSendSetYDScheduleRsp parameters

Parameter	Description
cluster	Cluster instance from which to send this command
dst	Requested destination address
rsp	Set Year Day Schedule response structure
callback	Callback function that is invoked when the response is received
arg	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.10.2.52 *ZbZclDoorLockServerSendToggleRsp*

```
enum ZclStatusCodeT ZbZclDoorLockServerSendToggleRsp(struct ZbZclClusterT *cluster, struct
ZbZclAddrInfoT *dst, struct ZbZclDoorLockToggleRspT *rsp, void (*callback)(struct
ZbApsdeDataConfT *conf, void *arg), void *arg);
```

Send a Toggle response command

Table 159. ZbZclDoorLockServerSendToggleRsp parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`rsp`	Toggle response structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.10.2.53 *ZbZclDoorLockServerSendUnlockRsp*

```
enum ZclStatusCodeT ZbZclDoorLockServerSendUnlockRsp(struct ZbZclClusterT *cluster, struct
ZbZclAddrInfoT *dst, struct ZbZclDoorLockUnlockDoorRspT *rsp, void (*callback)(struct
ZbApsdeDataConfT *conf, void *arg), void *arg);
```

Send an Unlock Door response command

Table 160. ZbZclDoorLockServerSendUnlockRsp parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`rsp`	Unlock Door response structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.10.2.54 *ZbZclDoorLockServerSendUnlockTimeoutRsp*

```
enum ZclStatusCodeT ZbZclDoorLockServerSendUnlockTimeoutRsp(struct ZbZclClusterT *cluster,
struct ZbZclAddrInfoT *dst, struct ZbZclDoorLockUnlockTimeoutRspT *rsp, void (*callback)
(struct ZbApsdeDataConfT *conf, void *arg), void *arg);
```

Send an Unlock with Timeout response command

Table 161. ZbZclDoorLockServerSendUnlockTimeoutRsp parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`rsp`	Unlock with Timeout response structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.10.3 Door lock cluster structures

5.10.3.1 *ZbZclDoorLockClrAllPinRspT*

Clear All PIN Codes response structure

Table 162. ZbZclDoorLockClrAllPinRspT parameters

Parameter	Description
`uint8_t status`	Status

5.10.3.2 *ZbZclDoorLockClrAllRfidRspT*

Clear All RFID Codes response structure

Table 163. ZbZclDoorLockClrAllRfidRspT parameters

Parameter	Dimension
`uint8_t status`	Status

5.10.3.3 *ZbZclDoorLockClrHDScheduleReqT*

Clear Holiday Schedule request structure

Table 164. ZbZclDoorLockClrHDScheduleReqT parameters

Parameter	Description
`uint8_t schedule_id`	Holiday schedule ID

5.10.3.4 *ZbZclDoorLockClrHDScheduleRspT*

Clear Holiday Schedule response structure

Table 165. ZbZclDoorLockClrHDScheduleRspT parameters

Parameter	Description
`uint8_t status`	Status

5.10.3.5 *ZbZclDoorLockClrPinReqT*

Clear PIN Code request structure

Table 166. ZbZclDoorLockClrPinReqT parameters

Parameter	Description
`uint16_t user_id`	User ID

5.10.3.6 *ZbZclDoorLockClrPinRspT*

Clear PIN Code response structure

Table 167. ZbZclDoorLockClrPinRspT parameters

Parameter	Description
`uint8_t status`	Status

5.10.3.7 *ZbZclDoorLockClrRfidReqT*

Clear RFID Code request structure

Table 168. ZbZclDoorLockClrRfidReqT parameters

Paramter	Description
`uint16_t user_id`	User ID

5.10.3.8 *ZbZclDoorLockClrRfidRspT*

Clear RFID Code response structure

Table 169. ZbZclDoorLockClrRfidRspT parameters

Parameter	Description
`uint8_t status`	Status

5.10.3.9 *ZbZclDoorLockClrWDScheduleReqT*

Clear Weekday Schedule request structure

Table 170. ZbZclDoorLockClrWDScheduleReqT parameters

Paramter	Description
`uint8_t schedule_id`	Schedule ID
`uint16_t user_id`	User ID

5.10.3.10 *ZbZclDoorLockClrWDScheduleRspT*

Clear Weekday Schedule response structure

Table 171. ZbZclDoorLockClrWDScheduleRspT parameters

Parameter	Description
`uint8_t status`	Status

5.10.3.11 *ZbZclDoorLockClrYDScheduleReqT*

Clear Year Day Schedule request structure

Table 172. ZbZclDoorLockClrYDScheduleReqT parameters

Parameter	Desciption
`uint8_t schedule_id`	Schedule ID
`uint16_t user_id`	User ID

5.10.3.12 *ZbZclDoorLockClrYDScheduleRspT*

Clear Year Day Schedule response structure

Table 173. ZbZclDoorLockClrYDScheduleRspT parameters

Parameter	Description
`uint8_t status`	Status

5.10.3.13 *ZbZclDoorLockGetHDScheduleReqT*

Get Holiday Schedule request structure

Table 174. ZbZclDoorLockGetHDScheduleReqT Parameters

Parameter	Description
`uint8_t schedule_id`	Holiday schedule ID

5.10.3.14 *ZbZclDoorLockGetHDScheduleRspT*

Get Holiday Schedule response structure

Table 175. ZbZclDoorLockGetHDScheduleRspT parameters

Parameter	Description
`uint8_t schedule_id`	Holiday schedule ID
`uint8_t status`	Status
`uint32_t local_start_time`	Local start time
`uint32_t local_end_time`	Local end time
`uint8_t operating_mode`	Operating mode during holiday

5.10.3.15 *ZbZclDoorLockGetLogReqT*

Get Log Record request structure

Table 176. ZbZclDoorLockGetLogReqT parameters

Parameter	Desciption
`uint16_t log_index`	Log index

5.10.3.16 *ZbZclDoorLockGetLogRspT*

Get Log Record response structure

Table 177. ZbZclDoorLockGetLogRspT parameters

Parameter	Description
`uint16_t log_entry_id`	Log entry ID
`uint32_t time_stamp`	Timestamp
`uint8_t event_type`	Event type
`uint8_t source`	Source (see Operation Event Sources)
`uint8_t alarm_code`	Event ID/Alarm Code (see Operation Event Codes)
`uint16_t user_id`	User ID
`uint8_t pin`	PIN
`uint8_t pin_len`	Length of PIN

5.10.3.17 *ZbZclDoorLockGetPinReqT*

Get PIN Code request structure

Table 178. ZbZclDoorLockGetPinReqT parameters

Parameter	Description
`uint16_t user_id`	User ID

5.10.3.18 *ZbZclDoorLockGetPinRspT*

Get PIN Code response structure

Table 179. ZbZclDoorLockGetPinRspT parameters

Parameter	Description
`uint16_t user_id`	User ID
`uint8_t user_status`	User status
`uint8_t user_type`	User type
`uint8_t pin`	PIN
`uint8_t pin_len`	Length of PIN

5.10.3.19 *ZbZclDoorLockGetRfidReqT*

Get RFID Code request structure

Table 180. ZbZclDoorLockGetRfidReqT parameters

Parameter	Description
`uint16_t user_id`	User ID

5.10.3.20 *ZbZclDoorLockGetRfidRspT*

Get RFID Code response structure

Table 181. ZbZclDoorLockGetRfidRspT parameters

Parameter	Description
`uint16_t user_id`	User ID
`uint8_t user_status`	User status
`uint8_t user_type`	User type
`uint8_t rfid`	RFID code
`uint8_t rfid_len`	Length of RFID code

5.10.3.21 *ZbZclDoorLockGetUserStatusReqT*

Get User Status request structure

Table 182. ZbZclDoorLockGetUserStatusReqT parameters

Parameter	Description
`uint16_t user_id`	User ID

5.10.3.22 *ZbZclDoorLockGetUserStatusRspT*

Get User Status response structure

Table 183. ZbZclDoorLockGetUserStatusRspT parameters

Parameter	Description
`uint16_t user_id`	User ID
`uint8_t user_status`	User status

5.10.3.23 *ZbZclDoorLockGetUserTypeReqT*

Get User Type request structure

Table 184. ZbZclDoorLockGetUserTypeReqT parameters

Parameter	Description
`uint16_t user_id`	User ID

5.10.3.24 *ZbZclDoorLockGetUserTypeRspT*

Get User Type response structure

Table 185. ZbZclDoorLockGetUserTypeRspT parameters

Parameter	Description
`uint16_t user_id`	User ID
`uint8_t user_type`	User type

5.10.3.25 *ZbZclDoorLockGetWDScheduleReqT*

Get Weekday Schedule request structure

Table 186. ZbZclDoorLockGetWDScheduleReqT parameters

ParameteR	Description
`uint8_t schedule_id`	Schedule ID
`uint16_t user_id`	User ID

5.10.3.26 *ZbZclDoorLockGetWDScheduleRspT*

Get Weekday Schedule response structure

Table 187. ZbZclDoorLockGetWDScheduleRspT parameters

Parameter	Description
`uint8_t schedule_id`	Schedule ID
`uint16_t user_id`	User ID
`uint8_t status`	Status
`uint8_t days_mask`	Days mask
`uint8_t start_hour`	Start hour
`uint8_t start_minute`	Start minute
`uint8_t end_hour`	End hour
`uint8_t end_minute`	End minute

5.10.3.27 *ZbZclDoorLockGetYDScheduleReqT*

Get Year Day Schedule request structure

Table 188. ZbZclDoorLockGetYDScheduleReqT parameters

Parameter	Description
`uint8_t schedule_id`	Schedule ID
`uint16_t user_id`	User ID

5.10.3.28 *ZbZclDoorLockGetYDScheduleRspT*

Get Year Day Schedule response structure

Table 189. ZbZclDoorLockGetYDScheduleRspT parameters

Parameter	Description
`uint8_t schedule_id`	Schedule ID
`uint16_t user_id`	User ID
`uint8_t status`	Status
`uint32_t local_start_time`	Local start time
`uint32_t local_end_time`	Local end time

5.10.3.29 *ZbZclDoorLockLockDoorReqT*

Lock Door request structure

Table 190. ZbZclDoorLockLockDoorReqT Parameters

Parameter	Description
`uint8_t pin`	PIN/RFID Code
`uint8_t pin_len`	Length of PIN

5.10.3.30 *ZbZclDoorLockLockDoorRspT*

Lock Door response structure

Table 191. ZbZclDoorLockLockDoorRspT parameters

Parameter	Desciption
`uint8_t status`	Status

5.10.3.31 *ZbZclDoorLockServerCallbacksT*

Door Lock Server callbacks configuration

Table 192. ZbZclDoorLockServerCallbacksT parameters

Parameter	Description
`lock` (callback function pointer)	`Enum ZclStatusCodeT (*lock)(struct ZbZclClusterT*cluster, struct ZbZclDoorLockLockDoorReqT *req, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of Lock Door command.

Parameter	Description
`unlock` (callback function pointer)	`Enum ZclStatusCodeT(*unlock)(struct ZbZclClusterT*cluster,struct ZbZclDoorLockUnlockDoorReqT *req, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of Unlock Door command.
`toggle` (callback function pointer)	`enumZclStatusCodeT (*toggle)(struct ZbZclClusterT*cluster, struct ZbZclDoorLockToggleReqT *req, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of Toggle command.
`unlock_timeout` (callback function pointer)	`enum ZclStatusCodeT (*unlock_timeout)(struct ZbZclClusterT *cluster,struct ZbZclDoorLockUnlockTimeoutReqT *req, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of Unlock with Timeout command.
`get_log` (callback function pointer)	`Enum ZclStatusCodeT (*get_log)(struct ZbZclClusterT*cluster, struct ZbZclDoorLockGetLogReqT *req, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of Get Log Record command.
`set_pin` (callback function pointer)	`Enum ZclStatusCodeT (*set_pin)(struct ZbZclClusterT*cluster, struct ZbZclDoorLockSetPinReqT *req, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of Set PIN Code command.
`get_pin` (callback function pointer)	`Enum ZclStatusCodeT (*get_pin)(struct ZbZclClusterT*cluster, struct ZbZclDoorLockGetPinReqT *req, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of Get PIN Code command.
`clr_pin` (callback function pointer)	`Enum ZclStatusCodeT (*clr_pin)(struct ZbZclClusterT*cluster, struct ZbZclDoorLockClrPinReqT *req, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of Clear PIN Code command.
`clr_all_pins` (callback function pointer)	`enum ZclStatusCodeT (*clr_all_pins)(struct ZbZclClusterT*cluster, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of Clear All PIN Codes command.
`set_user_status` (callback function pointer)	`enum ZclStatusCodeT (*set_user_status)(struct ZbZclClusterT *cluster, struct ZbZclDoorLockSetUserStatusReqT *req, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of Set User Status command.
`get_user_status` (callback function pointer)	`enum ZclStatusCodeT (*get_user_status)(struct ZbZclClusterT *cluster, struct ZbZclDoorLockGetUserStatusReqT *req, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of Get User Status command.
`set_wd_sched` (callback function pointer)	`enum ZclStatusCodeT (*set_wd_sched)(struct ZbZclClusterT*cluster,struct ZbZclDoorLockSetWDScheduleReqT *req, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of Set Weekday Schedule command.
`get_wd_sched` (callback function pointer)	`enum ZclStatusCodeT (*get_wd_sched)(struct ZbZclClusterT*cluster,struct ZbZclDoorLockGetWDScheduleReqT *req, struct ZbZclAddrInfoT *srcInfo, void *arg)`

Parameter	Description
	Callback to application, invoked on receipt of Get Weekday Schedule command.
`clr_wd_sched` (callback function pointer)	`enum ZclStatusCodeT (*clr_wd_sched)(struct ZbZclClusterT*cluster,struct ZbZclDoorLockClrWDScheduleReqT *req, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of Clear Weekday Schedule command.
`set_yd_sched` (callback function pointer)	`enum ZclStatusCodeT (*set_yd_sched)(struct ZbZclClusterT*cluster,struct ZbZclDoorLockSetYDScheduleReqT *req, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of Set Year Day Schedule command.
`get_yd_sched` (callback function pointer)	`enum ZclStatusCodeT (*get_yd_sched)(struct ZbZclClusterT*cluster,struct ZbZclDoorLockGetYDScheduleReqT *req, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of Get Year Day Schedule command.
`clr_yd_sched` (callback function pointer)	`enum ZclStatusCodeT (*clr_yd_sched)(struct ZbZclClusterT*cluster,struct ZbZclDoorLockClrYDScheduleReqT *req, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of Clear Year Day Schedule command.
`set_hd_sched` (callback function pointer)	`enum ZclStatusCodeT (*set_hd_sched)(struct ZbZclClusterT*cluster,struct ZbZclDoorLockSetHDScheduleReqT *req, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of Set Holiday Schedule command.
`get_hd_sched` (callback function pointer)	`enum ZclStatusCodeT (*get_hd_sched)(struct ZbZclClusterT*cluster,struct ZbZclDoorLockGetHDScheduleReqT *req, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of Get Holiday Schedule command.
`clr_hd_sched` (callback function pointer)	`enum ZclStatusCodeT (*clr_hd_sched)(struct ZbZclClusterT*cluster,struct ZbZclDoorLockClrHDScheduleReqT *req, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of Clear Holiday Schedule command.
`set_user_type` (callback function pointer)	`enum ZclStatusCodeT (*set_user_type)(struct ZbZclClusterT *cluster,struct ZbZclDoorLockSetUserTypeReqT *req, struct ZbZclAddrInfoT*srcInfo, void *arg)` Callback to application, invoked on receipt of Set User Type command.
`get_user_type` (callback function pointer)	`enum ZclStatusCodeT (*get_user_type)(struct ZbZclClusterT *cluster, struct ZbZclDoorLockGetUserTypeReqT *req, struct ZbZclAddrInfoT*srcInfo, void *arg)` Callback to application, invoked on receipt of Get User Type command.
`set_rfid` (callback function pointer)	`Enum ZclStatusCodeT (*set_rfid)(struct ZbZclClusterT*cluster, struct ZbZclDoorLockSetRfidReqT *req, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of Set RFID Code command.
`get_rfid` (callback function pointer)	`Enum ZclStatusCodeT (*get_rfid)(struct ZbZclClusterT*cluster, struct ZbZclDoorLockGetRfidReqT *req, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of Get RFID Code command.

Parameter	Description
`clr_rfid` (callback function pointer)	`Enum ZclStatusCodeT (*clr_rfid)(struct ZbZclClusterT*cluster, struct ZbZclDoorLockClrRfidReqT *req, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of Clear RFID Code command.
`clr_all_rfids` (callback function pointer)	`enum ZclStatusCodeT (*clr_all_rfids)(struct ZbZclClusterT *cluster, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of Clear All RFID Codes command.

5.10.3.32 *ZbZclDoorLockSetHDScheduleReqT*

Set Holiday Schedule request structure

Table 193. ZbZclDoorLockSetHDScheduleReqT Parameters

Parameter	Description
`uint8_t schedule_id`	Holiday schedule ID
`uint32_t local_start_time`	Local start time
`uint32_t local_end_time`	Local end time
`uint8_t operating_mode`	Operating mode during holiday

5.10.3.33 *ZbZclDoorLockSetHDScheduleRspT*

Set Holiday Schedule response structure

Table 194. ZbZclDoorLockSetHDScheduleRspT parameters

Parameter	Description
`uint8_t status`	Status

5.10.3.34 *ZbZclDoorLockSetPinReqT*

Set PIN Code request structure

Table 195. ZbZclDoorLockSetPinReqT parameters

Parameter	Description
`uint16_t user_id`	User ID
`uint8_t user_status`	User status
`uint8_t user_type`	User type
`uint8_t pin`	PIN
`uint8_t pin_len`	Length of PIN

5.10.3.35 *ZbZclDoorLockSetPinRspT*

Set PIN Code response structure

Table 196. ZbZclDoorLockSetPinRspT parameters

Parameter	Description
`uint8_t status`	Status

5.10.3.36 *ZbZclDoorLockSetRfidReqT*

Set RFID Code request structure

Table 197. ZbZclDoorLockSetRfidReqT parameters

Parameter	Description
`uint16_t user_id`	User ID
`uint8_t user_status`	User status
`uint8_t user_type`	User type
`uint8_t` **`rfid`**	PIN/RFID Code
`uint8_t rfid_len`	Length of RFID code

5.10.3.37 *ZbZclDoorLockSetRfidRspT*

Set RFID Code response structure

Table 198. ZbZclDoorLockSetRfidRspT parameters

Paramter	Description
`uint8_t status`	Status

5.10.3.38 *ZbZclDoorLockSetUserStatusReqT*

Set User Status request structure

Table 199. ZbZclDoorLockSetUserStatusReqT Parameters

Parameter	Description
`uint16_t user_id`	User ID
`uint8_t user_status`	User status

5.10.3.39 *ZbZclDoorLockSetUserStatusRspT*

Set User Status response structure

Table 200. ZbZclDoorLockSetUserStatusRspT parameters

Parameter	Description
`uint8_t status`	Status

5.10.3.40 *ZbZclDoorLockSetUserTypeReqT*

Set User Type request structure

Table 201. ZbZclDoorLockSetUserTypeReqT parameters

Paramter	Description
`uint16_t user_id`	User ID
`uint8_t user_type`	User type

5.10.3.41 *ZbZclDoorLockSetUserTypeRspT*

Set User Type response structure

Table 202. ZbZclDoorLockSetUserTypeRspT Parameters

Parameter	Description
`uint8_t status`	Status

5.10.3.42 *ZbZclDoorLockSetWDScheduleReqT*

Set Weekday Schedule request structure

Table 203. ZbZclDoorLockSetWDScheduleReqT parameters

Parameter	Description
`uint8_t schedule_id`	Schedule ID number
`uint16_t user_id`	User ID
`uint8_t days_mask`	Days mask
`uint8_t start_hour`	Start hour
`uint8_t start_minute`	Start minute
`uint8_t end_hour`	End hour
`uint8_t end_minute`	End minute

5.10.3.43 *ZbZclDoorLockSetWDScheduleRspT*

Set Weekday Schedule response structure

Table 204. ZbZclDoorLockSetWDScheduleRspT parameters

Parameter	Description
`uint8_t status`	Status

5.10.3.44 *ZbZclDoorLockSetYDScheduleReqT*

Set Year Day Schedule request structure

Table 205. ZbZclDoorLockSetYDScheduleReqT parameters

Parameter	Description
`uint8_t schedule_id`	Schedule ID
`uint16_t user_id`	User ID
`uint32_t local_start_time`	Local start time
`uint32_t local_end_time`	Local end time

5.10.3.45 *ZbZclDoorLockSetYDScheduleRspT*

Set Year Day Schedule response structure

Table 206. ZbZclDoorLockSetYDScheduleRspT parameters

Parameter	Description
`uint8_t status`	Status

5.10.3.46 *ZbZclDoorLockToggleReqT*

Toggle request structure

Table 207. ZbZclDoorLockToggleReqT parameters

Parameter	Description
`uint8_t pin`	PIN/RFID Code
`uint8_t pin_len`	Length of PIN/RFID Code

5.10.3.47 *ZbZclDoorLockToggleRspT*

Toggle response structure

Table 208. ZbZclDoorLockToggleRspT parameters

Parameter	Description
`uint8_t status`	Status

5.10.3.48 *ZbZclDoorLockUnlockDoorReqT*

Unlock Door request structure

Table 209. ZbZclDoorLockUnlockDoorReqT parameters

Parameter	Description
`uint8_t pin`	PIN/RFID Code
`uint8_t pin_len`	Length of PIN/RFID Code

5.10.3.49 *ZbZclDoorLockUnlockDoorRspT*

Unlock Door response structure

Table 210. ZbZclDoorLockUnlockDoorRspT parameters

Parameter	Description
`uint8_t status`	Status

5.10.3.50 *ZbZclDoorLockUnlockTimeoutReqT*

Unlock with Timeout request structure

Table 211. ZbZclDoorLockUnlockTimeoutReqT parameters

Parameter	Description
`uint16_t timeout`	Timeout in seconds
`uint8_t pin`	PIN/RFID Code
`uint8_t pin_len`	Length of PIN/RFID Code

5.10.3.51 *ZbZclDoorLockUnlockTimeoutRspT*

Unlock with Timeout response structure

Table 212. ZbZclDoorLockUnlockTimeoutRspT parameters

Parameter	Description
`uint8_t status`	Status

5.11 Demand response and load control cluster

```
#include "zcl/se/zcl.drlc.h"
```

5.11.1 Demand response and load control cluster description

Table 213. Demand response and load control cluster PICS code description

PICS code	Name	Supported	Notes
DRLC.S	Server	True	-
DRLC.C	Client	True	-
DRLC.S.Afffd	ClusterRevision server attribute	True	-
DRLC.S.Afffe	AttributeReportingStatus server attribute	False	-
DRLC.S.C00.Rsp	Report Event Status server command	True	-
DRLC.S.C01.Rsp	Get Scheduled Events server command	True	-
DRLC.S.C00.Tx	Load Control Event server command	True	-
DRLC.S.C01.Tx	Cancel Load Control Event server command	True	-
DRLC.S.C02.Tx	Cancel All Load Control Events server command	True	-
DRLC.C.A0000	UtilityEnrollmentGroup client attribute	True	-
DRLC.C.A0001	StartRandomizeMinutes client attribute	True	-
DRLC.C.A0002	StopRandomizeMinutes client attribute	True	-
DRLC.C.A0003	DeviceClassValue client attribute	True	-
DRLC.C.Afffd	ClusterRevision client attribute	True	-
DRLC.C.Afffe	AttributeReportingStatus client attribute	False	-
DRLC.C.C00.Rsp	Load Control Event client command	True	-
DRLC.C.C01.Rsp	Cancel Load Control Event client command	True	-
DRLC.C.C02.Rsp	Cancel All Load Control Events client command	True	-
DRLC.C.C00.Tx	Report Event Status client command	True	-
DRLC.C.C01.Tx	Get Scheduled Events client command	True	-

5.11.2 Demand response and load control cluster functions

5.11.2.1 *ZbZclDrlcClientAlloc*

```
struct ZbZclClusterT * ZbZclDrlcClientAlloc(struct ZigBeeT *zb, uint8_t endpoint, struct
ZbZclClusterT *time_server, struct ZbZclDrlcClientCallbacksT *callbacks, void *cb_arg);
```

Create a new instance of the DRLC Client cluster

Table 214. ZbZclDrlcClientAlloc parameters

Parameter	Description
zb	Zigbee® stack instance stack instance
endpoint	Endpoint on which to create cluster
time_server	Time server cluster instance used to retrieve event timing information
callbacks	Structure containing any callback function pointers for this cluster
cb_arg	Pointer to application data that is provided back to the callback function when invoked

Return: Cluster pointer, or NULL if there is an error.

5.11.2.2 *ZbZclDrlcClientCommandGetEventsReq*

```
enum ZclStatusCodeT ZbZclDrlcClientCommandGetEventsReq(struct ZbZclClusterT*cluster, const
struct ZbApsAddrT *dst, struct ZbZclDrlcGetEventsReqT *cmd_req, void (*callback)(struct
ZbZclCommandRspT *zcl_rsp, void *arg), void *arg);
```

Send a Get Scheduled Events

Table 215. ZbZclDrlcClientCommandGetEventsReq parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`cmd_req`	Get Scheduled Events command request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.11.2.3 *ZbZclDrlcClientCommandReportStatusReq*

```
enum ZclStatusCodeT ZbZclDrlcClientCommandReportStatusReq(struct ZbZclClusterT*cluster,
const struct ZbApsAddrT *dst, struct ZbZclDrlcStatusT *statusPtr, void (*callback)(struct
ZbZclCommandRspT *zcl_rsp, void *arg), void *arg);
```

Send a Report Event Status Command

Table 216. ZbZclDrlcClientCommandReportStatusReq Parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`statusPtr`	Report Event Status command request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.11.2.4 *ZbZclDrlcClientGetEventList*

```
unsigned int ZbZclDrlcClientGetEventList(struct ZbZclClusterT *cluster, struct
ZbZclDrlcEventT *eventList, unsigned int maxEntries);
```

Send a Get Event List command

Table 217. ZbZclDrlcClientGetEventList parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`eventList`	Holds a pointer to the event list if successful
`maxEntries`	Max event list entries

Return: Event list size if successful, or 0 on error

5.11.2.5 ZbZclDrlcServerAlloc

```
struct ZbZclClusterT * ZbZclDrlcServerAlloc(struct ZigBeeT *zb, uint8_t endpoint, struct
ZbZclDrlcServerCallbacksT *callbacks, void *arg);
```

Create a new instance of the DRLC Server cluster

Table 218. ZbZclDrlcServerAlloc parameters

Parameter	Description
zb	Zigbee® stack instance
endpoint	Endpoint on which to create cluster
callbacks	Structure containing any callback function pointers for this cluster
arg	Pointer to application data that is provided back to the callback function when invoked

Return: Cluster pointer, or NULL if there is an error.

5.11.2.6 ZbZclDrlcServerCommandCancelAllReq

```
enum ZclStatusCodeT ZbZclDrlcServerCommandCancelAllReq(struct ZbZclClusterT*cluster, uint8_t
ctrl, const struct ZbApsAddrT *dst, void (*callback)(struct ZbZclCommandRspT *zcl_rsp, void
*arg), void *arg);
```

Send a Cancel All Load Control Event command

Table 219. ZbZclDrlcServerCommandCancelAllReq parameters

Parameter	Description
cluster	Cluster instance from which to send this command
ctrl	Cancel control bitmap, used to determine cancellation method, for examole ZCL_DRLC_CANCEL_CONTROL_GRACEFUL
dst	Requested destination address
callback	Callback function that is invoked when the response is received
arg	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.11.2.7 ZbZclDrlcServerCommandCancelReq

```
enum ZclStatusCodeT ZbZclDrlcServerCommandCancelReq(struct ZbZclClusterT *cluster, struct
ZbZclDrlcCancelT *cancelInfoPtr, const struct ZbApsAddrT *dst, void (*callback)(struct
ZbZclCommandRspT *zcl_rsp, void *arg), void *arg);
```

Send a Cancel Load Control Event command

Table 220. ZbZclDrlcServerCommandCancelReq parameters

Parameter	Description
cluster	Cluster instance from which to send this command
eventPtr	Cancel Load Control Event command request structure
dst	Requested destination address
callback	Callback function that is invoked when the response is received

Parameter	Description
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.11.2.8 *ZbZclDrlcServerCommandEventReq*

```
enum ZclStatusCodeT ZbZclDrlcServerCommandEventReq(struct ZbZclClusterT *cluster, struct
ZbZclDrlcEventT *eventPtr, const struct ZbApsAddrT *dst, void (*callback)(struct
ZbZclCommandRspT *zcl_rsp, void *arg), void *arg);
```

Send a Load Control Event command

Table 221. **ZbZclDrlcServerCommandEventReq parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`eventPtr`	Load Control Event command request structure
`dst`	Requested destination address
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.11.3 Demand response and load control cluster enumerations

5.11.3.1 *ZbZclDrlcCliAttrT*

DRLC client attribute IDs

Table 222. **ZbZclDrlcCliAttrT attributes description**

Attribute	Description
ZCL_DRLC_CLI_ATTR_UTILITY_ENROL_GRP	`UtilityEnrollmentGroup`
ZCL_DRLC_CLI_ATTR_START_RAND_MINS	`StartRandomizationMinutes`
ZCL_DRLC_CLI_ATTR_DURATION_RAND_MINS	`DurationRandomizationMinutes`
ZCL_DRLC_CLI_ATTR_DEVICE_CLASS	`DeviceClassValue`

5.11.4 Demand response and load control cluster structures

5.11.4.1 *ZbZclDrlcCancelT*

Cancel Load Control Event command structure

Table 223. **ZbZclDrlcCancelT parameters**

Parameter	Description
`uint32_t issuer_event_id`	Issuer event ID
`uint16_t device_class`	Device class
`uint8_t util_enrol_group`	Utility Enrollment Group
`uint8_t cancel_control`	Cancel Control - for example ZCL_DRLC_CANCEL_CONTROL_GRACEFUL

5.11.4.2 *ZbZclDrlcClientCallbacksT*

DRLC Client callbacks configuration

Table 224. ZbZclDrlcClientCallbacksT parameters

Parameter	Description
`start` (callback function pointer)	`bool (*start)(void *arg, struct ZbZclDrlcEventT *event)` Callback to application, invoked on the start of an event
`stop` (callback function pointer)	`void (*stop)(void *arg, struct ZbZclDrlcEventT *event)` Callback to application, invoked on the end of an event

5.11.4.3 *ZbZclDrlcEventT*

Load Control Event command structure

Table 225. ZbZclDrlcEventT parameters

Parameter	Description
`uint32_t issuer_id`	Issuer event ID
`uint16_t device_class`	Device class
`uint8_t util_enrol_group`	Utility Enrollment Group
`uint32_t start_time`	Start time - UTC time, or 0 = now
`uint16_t duration`	Duration in minutes
`enum ZbZclDrlcCriticalityLevelTcriticality`	Criticality level
`uint8_t cool_offset`	Cooling temperature offset (optional)
`uint8_t heat_offset`	Heating temperature offset (optional)
`int16_t cool_setpoint`	Cooling temperature set point (optional)
`int16_t heat_setpoint`	Heating temperature set point (optional)
`int8_t avg_load_adj`	Average load adjustment percentage (optional)
uint8_t dutycycle	Duty cycle (optional)
`uint8_t event_control`	Event Control - for example ZCL_DRLC_EVENT_CTRL_RAND_START

5.11.4.4 *ZbZclDrlcGetEventsReqT*

Get Scheduled Events command structure

Table 226. ZbZclDrlcGetEventsReqT Parameters

Parameter	Description
`uint32_t start_time`	Start Time
`uint8_t num_events`	Number of Events
`uint32_t issuer_id`	Issuer Event ID (optional)

5.11.4.5 *ZbZclDrlcServerCallbacksT*

DRLC Server callbacks configuration

Table 227. ZbZclDrlcServerCallbacksT parameters

Parameter	Description
`report_status` (callback function pointer)	`Void (*report_status)(struct ZbZclClusterT *cluster, struct ZbZclAddrInfoT*srcInfo,struct ZbZclDrlcStatusT*status, void *arg)` Callback to application, invoked on receipt of Report Event Status command
`get_events` (callback function pointer)	`Enum ZclStatusCodeT (*get_events)(struct ZbZclClusterT*cluster, struct ZbZclAddrInfoT *srcInfo,struct ZbZclDrlcGetEventsReqT *req, void *arg)` Callback to application, invoked on receipt of Get Scheduled Events command. The `get_events` callback handler in the application must return ZCL_STATUS_SUCCESS, or ZCL_STATUS_NOT_FOUND if no events are found. Events are re-issued by calling `ZbZclDrlcServerCommandEventReq()`

5.11.4.6 *ZbZclDrlcStatusT*

Report Event Status command structure

Table 228. ZbZclDrlcStatusT parameters

Parameter	Description
`uint32_t issuer_id`	Issuer Event ID
`enum ZbZclDrlcEventStatusT status`	Event Status
`uint32_t status_time`	Event Status Time
`enum ZbZclDrlcCriticalityLevelTcrit_level_applied`	Criticality Level Applied
`uint16_t cool_setpoint_applied`	Cooling temperature set point applied (optional) - Set to ZCL_DRLC_COOL_SETPOINT_IGNORED if not used
`uint16_t heat_setpoint_applied`	Heating Temperature Set Point Applied (optional) – Set to ZCL_DRLC_HEAT_SETPOINT_IGNORED if not used
`int8_t avg_load_adj_applied`	Average Load Adjustment Percentage Applied (optional) - Set to ZCL_DRLC_AVG_LOAD_ADJ_IGNORED if not used
`uint8_t dutycycle_applied`	Duty Cycle Applied (optional) - Set to ZCL_DRLC_DUTYCYCLE_IGNORED if not used
`uint8_t event_control`	Event Control
`enum ZbZclDrlcSignatureT sig_type`	Signature Type
`uint8_t sig_data`	Signature (optional)

5.12 Electrical measurement cluster

```
#include "zcl/general/zcl.elec.meas.h"
```

5.12.1 Electrical measurement cluster description

Table 229. Electrical measurement cluster PICS code description

PICS Code	Name	Supported	Notes
EMR.S	Server	True	-
EMR.C	Client	True	-
EMR.S.A0000	`MeasurementType` server attribute	True	-

PICS Code	Name	Supported	Notes
EMR.S.A0100	DCVoltage server attribute	True	Optional
EMR.S.A0101	DCVoltageMin server attribute	True	Optional
EMR.S.A0102	DCVoltageMax server attribute	True	Optional
EMR.S.A0200	DCVoltageMultiplier server attribute	True	Optional
EMR.S.A0201	DCVoltageDivisor server attribute	True	Optional
EMR.S.A0700	DCOverloadAlarmsMask server attribute	True	Optional
EMR.S.A0701	DCVoltageOverload server attribute	True	Optional
EMR.S.Afffd	ClusterRevision server attribute	True	-
EMR.S.Afffe	AttributeReportingStatus server attribute	False	-
EMR.S.C00.Rsp	Get Profile Info Command server command	Depends on customer application	-
EMR.S.C01.Rsp	Get Measurement Profile Command server command	Depends on customer application	-
EMR.S.C00.Tx	Get Profile Info Response Command server command	Depends on customer application	-
EMR.S.C01.Tx	Get Measurement Profile Response Command server command	Depends on customer application	-
DRLK.C.Afffd	ClusterRevision client attribute	True	-
DRLK.C.Afffe	AttributeReportingStatus client attribute	False	-
EMR.C.C00.Rsp	Get Profile Info Response Command client command	Depends on customer application	-
EMR.C.C01.Rsp	Get Measurement Profile Response Command client command	Depends on customer application	-
EMR.C.C00.Tx	Get Profile Info Command client command	True	-
EMR.C.C01.Tx	Get Measurement Profile Command client command	True	-

5.12.2 Electrical measurement cluster functions

5.12.2.1 *ZbZclElecMeasClientAlloc*

```
struct ZbZclClusterT * ZbZclElecMeasClientAlloc(struct ZigBeeT *zb, uint8_t endpoint);
```

Create a new instance of the Electrical Measurement Client cluster

Table 230. **ZbZclElecMeasClientAlloc parameters**

Parameter	Description
zb	Zigbee® stack instance
endpoint	Endpoint on which to create cluster

Return: Cluster pointer, or NULL if there is an error.

5.12.2.2 *ZbZclElecMeasClientGetMeasProfileReq*

```
enum ZclStatusCodeT ZbZclElecMeasClientGetMeasProfileReq(struct ZbZclClusterT*cluster, const
struct ZbApsAddrT *dst, struct ZbZclElecMeasClientGetMeasProfileReqT *req, void (*callback)
(struct ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Get Measurement Profile command

Table 231. ZbZclElecMeasClientGetMeasProfileReparameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`req`	Get Measurement Profile command structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.12.2.3 *ZbZclElecMeasClientGetProfileInfoReq*

```
enum ZclStatusCodeT ZbZclElecMeasClientGetProfileInfoReq(struct ZbZclClusterT*cluster, const
struct ZbApsAddrT *dst, void (*callback)(struct ZbZclCommandRspT*rsp, void *arg), void *arg);
```

Send a Get Profile Info command

Table 232. ZbZclElecMeasClientGetProfileInfoReq parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.12.2.4 *ZbZclElecMeasServerAlloc*

```
struct ZbZclClusterT * ZbZclElecMeasServerAlloc(struct ZigBeeT *zb, uint8_t endpoint, struct
ZbZclElecMeasSvrCallbacksT *callbacks, void *arg);
```

Create a new instance of the Electrical Measurement Server cluster

Table 233. ZbZclElecMeasServerAlloc parameters

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster
`callbacks`	Structure containing any callback function pointers for this cluster
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: Cluster pointer, or NULL if there is an error.

5.12.2.5 *ZbZclElecMeasServerSendMeasProfileRsp*

```
enum ZclStatusCodeT ZbZclElecMeasServerSendMeasProfileRsp(struct ZbZclClusterT*cluster,
struct ZbZclAddrInfoT *dst, struct ZbZclElecMeasSvrGetMeasProfileRspT*rsp);
```

Send a Get Measurement Profile response

Table 234. ZbZclElecMeasServerSendMeasProfileRsp parameters

Parameter	Description
cluster	Cluster instance from which to send this command
dst	Requested destination address
rsp	Get Measurement Profile response structure

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.12.2.6 *ZbZclElecMeasServerSendProfileInfoRsp*

```
enum ZclStatusCodeT ZbZclElecMeasServerSendProfileInfoRsp(struct ZbZclClusterT*cluster,
struct ZbZclAddrInfoT *dst, struct ZbZclElecMeasSvrGetProfileInfoRspT*rsp);
```

Send a Get Profile Info response

Table 235. ZbZclElecMeasServerSendProfileInfoRsp parameters

Parameter	Description
cluster	Cluster instance from which to send this command
dst	Requested destination address
rsp	Get Profile Info response structure

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.12.3 Electrical measurement cluster enumerations

5.12.3.1 *ZbZclElecMeasSvrAttrT*

Electrical Measurement server attribute IDs

Table 236. attributes

Attribute	Description
ZCL_ELEC_MEAS_ATTR_MEAS_TYPE	Measurement Type
ZCL_ELEC_MEAS_ATTR_DC_VOLT	DC Voltage (optional)
ZCL_ELEC_MEAS_ATTR_DC_VOLT_MIN	DC Voltage Min (optional)
ZCL_ELEC_MEAS_ATTR_DC_VOLT_MAX	DC Voltage Max (optional)
ZCL_ELEC_MEAS_ATTR_DC_CURRENT	DC Current (optional)
ZCL_ELEC_MEAS_ATTR_DC_CURRENT_MIN	DC Current Min (optional)
ZCL_ELEC_MEAS_ATTR_DC_CURRENT_MAX	DC Current Max (optional)
ZCL_ELEC_MEAS_ATTR_DC_POWER	DC Power (optional)
ZCL_ELEC_MEAS_ATTR_DC_POWER_MIN	DC Power Min (optional)
ZCL_ELEC_MEAS_ATTR_DC_POWER_MAX	DC Power Max (optional)
ZCL_ELEC_MEAS_ATTR_DC_VOLT_MULTIPLIER	DC Voltage Multiplier (optional)
ZCL_ELEC_MEAS_ATTR_DC_VOLT_DIVISOR	DC Voltage Divisor (optional)
ZCL_ELEC_MEAS_ATTR_DC_CURR_MULTIPLIER	DC Current Multiplier (optional)
ZCL_ELEC_MEAS_ATTR_DC_CURR_DIVISOR	DC Current Divisor (optional)
ZCL_ELEC_MEAS_ATTR_DC_PWR_MULTIPLIER	DC Power Multiplier (optional)
ZCL_ELEC_MEAS_ATTR_DC_PWR_DIVISOR	DC Power Divisor (optional)

Attribute	Description
ZCL_ELEC_MEAS_ATTR_AC_FREQ	AC Frequency (optional)
ZCL_ELEC_MEAS_ATTR_AC_FREQ_MIN	AC Frequency Min (optional)
ZCL_ELEC_MEAS_ATTR_AC_FREQ_MAX	AC Frequency Max (optional)
ZCL_ELEC_MEAS_ATTR_NEUTRAL_CURR	Neutral Current (optional)
ZCL_ELEC_MEAS_ATTR_TOTAL_ACTIVE_PWR	Total Active Power (optional)
ZCL_ELEC_MEAS_ATTR_TOTAL_REACTIVE_PWR	Total Reactive Power (optional)
ZCL_ELEC_MEAS_ATTR_TOTAL_APPARENT_PWR	Total Apparent Power (optional)
ZCL_ELEC_MEAS_ATTR_1ST_HARM_CURR	Measured 1st Harmonic Current (optional)
ZCL_ELEC_MEAS_ATTR_3RD_HARM_CURR	Measured 3rd Harmonic Current (optional)
ZCL_ELEC_MEAS_ATTR_5TH_HARM_CURR	Measured 5th Harmonic Current (optional)
ZCL_ELEC_MEAS_ATTR_7TH_HARM_CURR	Measured 7th Harmonic Current (optional)
ZCL_ELEC_MEAS_ATTR_9TH_HARM_CURR	Measured 9th Harmonic Current (optional)
ZCL_ELEC_MEAS_ATTR_11TH_HARM_CURR	Measured 11th Harmonic Current (optional)
ZCL_ELEC_MEAS_ATTR_PHASE_1ST_HARM_CURR	Measured Phase 1st Harmonic Current (optional)
ZCL_ELEC_MEAS_ATTR_PHASE_3RD_HARM_CURR	Measured Phase 3rd Harmonic Current (optional)
ZCL_ELEC_MEAS_ATTR_PHASE_5TH_HARM_CURR	Measured Phase 5th Harmonic Current (optional)
ZCL_ELEC_MEAS_ATTR_PHASE_7TH_HARM_CURR	Measured Phase 7th Harmonic Current (optional)
ZCL_ELEC_MEAS_ATTR_PHASE_9TH_HARM_CURR	Measured Phase 9th Harmonic Current (optional)
ZCL_ELEC_MEAS_ATTR_PHASE_11TH_HARM_CURR	Measured Phase 11th Harmonic Current (optional)
ZCL_ELEC_MEAS_ATTR_AC_FREQ_MULTIPLIER	AC Frequency Multiplier (optional)
ZCL_ELEC_MEAS_ATTR_AC_FREQ_DIVISOR	AC Frequency Divisor (optional)
ZCL_ELEC_MEAS_ATTR_PWR_MULTIPLIER	Power Multiplier (optional)
ZCL_ELEC_MEAS_ATTR_PWR_DIVISOR	Power Divisor (optional)
ZCL_ELEC_MEAS_ATTR_HARM_CURR_MULTIPLIER	Harmonic Current Multiplier (optional)
ZCL_ELEC_MEAS_ATTR_PHASE_CURR_MULTIPLIER	Phase Harmonic Current Multiplier (optional)
ZCL_ELEC_MEAS_ATTR_LINE_CURR	Line Current (optional)
ZCL_ELEC_MEAS_ATTR_ACTIVE_CURR	Active Current (optional)
ZCL_ELEC_MEAS_ATTR_REACTIVE_CURR	Reactive Current (optional)
ZCL_ELEC_MEAS_ATTR_RMS_VOLT	RMS Voltage (optional)
ZCL_ELEC_MEAS_ATTR_RMS_VOLT_MIN	RMS Voltage Min (optional)
ZCL_ELEC_MEAS_ATTR_RMS_VOLT_MAX	RMS Voltage Max (optional)
ZCL_ELEC_MEAS_ATTR_RMS_CURR	RMS Current (optional)
ZCL_ELEC_MEAS_ATTR_RMS_CURR_MIN	RMS Current Min (optional)
ZCL_ELEC_MEAS_ATTR_RMS_CURR_MAX	RMS Current Max (optional)
ZCL_ELEC_MEAS_ATTR_ACTIVE_PWR	Active Power (optional)
ZCL_ELEC_MEAS_ATTR_ACTIVE_PWR_MIN	Active Power Min (optional)
ZCL_ELEC_MEAS_ATTR_ACTIVE_PWR_MAX	Active Power Max (optional)
ZCL_ELEC_MEAS_ATTR_REACTIVE_PWR	Reactive Power (optional)
ZCL_ELEC_MEAS_ATTR_APPARENT_PWR	Apparent Power (optional)
ZCL_ELEC_MEAS_ATTR_PWR_FACTOR	Power Factor (optional)
ZCL_ELEC_MEAS_ATTR_AVG_RMS_VOLT_PERIOD	Average RMS Voltage Measurement Period (optional)

Attribute	Description
ZCL_ELEC_MEAS_ATTR_AVG_RMS_OV_COUNT	Average RMS Over Voltage Counter (optional)
ZCL_ELEC_MEAS_ATTR_AVG_RMS_UV_COUNT	Average RMS Under Voltage Counter (optional)
ZCL_ELEC_MEAS_ATTR_RMS_EXT_OVER_PERIOD	RMS Extreme Over Voltage Period (optional)
ZCL_ELEC_MEAS_ATTR_RMS_EXT_UNDER_PERIOD	RMS Extreme Under Voltage Period (optional)
ZCL_ELEC_MEAS_ATTR_VOLT_SAG_PERIOD	RMS Voltage Sag Period (optional)
ZCL_ELEC_MEAS_ATTR_VOLT_SWELL_PERIOD	RMS Voltage Swell Period (optional)
ZCL_ELEC_MEAS_ATTR_AC_VOLT_MULTIPLIER	AC Voltage Multiplier (optional)
ZCL_ELEC_MEAS_ATTR_AC_VOLT_DIVISOR	AC Voltage Divisor (optional)
ZCL_ELEC_MEAS_ATTR_AC_CURR_MULT	AC Current Multiplier (optional)
ZCL_ELEC_MEAS_ATTR_AC_CURR_DIVISOR	AC Current Divisor (optional)
ZCL_ELEC_MEAS_ATTR_AC_PWR_MULTIPLIER	AC Power Multiplier (optional)
ZCL_ELEC_MEAS_ATTR_AC_PWR_DIVISOR	AC Power Divisor (optional)
ZCL_ELEC_MEAS_ATTR_DC_OL_ALARMS_MASK	DC Overload Alarms Mask (optional)
ZCL_ELEC_MEAS_ATTR_DC_VOLT_OL	DC Voltage Overload (optional)
ZCL_ELEC_MEAS_ATTR_DC_CURR_OL	DC Current Overload (optional)
ZCL_ELEC_MEAS_ATTR_AC_ALARMS_MASK	AC Alarms Mask (optional)
ZCL_ELEC_MEAS_ATTR_AC_VOLT_OL	AC Voltage Overload (optional)
ZCL_ELEC_MEAS_ATTR_AC_CURR_OL	AC Current Overload (optional)
ZCL_ELEC_MEAS_ATTR_AC_ACTIVE_PWR_OL	AC Active Power Overload (optional)
ZCL_ELEC_MEAS_ATTR_AC_REACTIVE_PWR_OL	AC Reactive Power Overload (optional)
ZCL_ELEC_MEAS_ATTR_AVG_RMS_OV	Average RMS Over Voltage (optional)
ZCL_ELEC_MEAS_ATTR_AVG_RMS_UV	Average RMS Under Voltage (optional)
ZCL_ELEC_MEAS_ATTR_RMS_EXT_OV	RMS Extreme Over Voltage (optional)
ZCL_ELEC_MEAS_ATTR_RMS_EXT_UV	RMS Extreme Under Voltage (optional)
ZCL_ELEC_MEAS_ATTR_RMS_VOLT_SAG	RMS Voltage Sag (optional)
ZCL_ELEC_MEAS_ATTR_RMS_VOLT_SWELL	RMS Voltage Swell (optional)
ZCL_ELEC_MEAS_ATTR_LINE_CURR_B	Line Current PhB (optional)
ZCL_ELEC_MEAS_ATTR_ACTIVE_CURR_B	Active Current PhB (optional)
ZCL_ELEC_MEAS_ATTR_REACTIVE_CURR_B	Reactive Current PhB (optional)
ZCL_ELEC_MEAS_ATTR_RMS_VOLT_B	RMS Voltage PhB (optional)
ZCL_ELEC_MEAS_ATTR_RMS_VOLT_MIN_B	RMS Voltage Min PhB (optional)
ZCL_ELEC_MEAS_ATTR_RMS_VOLT_MAX_B	RMS Voltage Max PhB (optional)
ZCL_ELEC_MEAS_ATTR_RMS_CURR_B	RMS Current PhB (optional)
ZCL_ELEC_MEAS_ATTR_RMS_CURR_MIN_B	RMS Current Min PhB (optional)
ZCL_ELEC_MEAS_ATTR_RMS_CURR_MAX_B	RMS Current Max PhB (optional)
ZCL_ELEC_MEAS_ATTR_ACTIVE_PWR_B	Active Power PhB (optional)
ZCL_ELEC_MEAS_ATTR_ACTIVE_PWR_MIN_B	Active Power Min PhB (optional)
ZCL_ELEC_MEAS_ATTR_ACTIVE_PWR_MAX_B	Active Power Max PhB (optional)
ZCL_ELEC_MEAS_ATTR_REACTIVE_PWR_B	Reactive Power PhB (optional)
ZCL_ELEC_MEAS_ATTR_APPARENT_PWR_B	Apparent Power PhB (optional)
ZCL_ELEC_MEAS_ATTR_PWR_FACTOR_B	Power Factor PhB (optional)

Attribute	Description
ZCL_ELEC_MEAS_ATTR_AVG_RMS_VOLT_PERIOD_ B	Average RMS Voltage Measurement Period PhB (optional)
ZCL_ELEC_MEAS_ATTR_AVG_RMS_OV_B	Average RMS Over Voltage Counter PhB (optional)
ZCL_ELEC_MEAS_ATTR_AVG_RMS_UV_B	Average RMS Under Voltage Counter PhB (optional)
ZCL_ELEC_MEAS_ATTR_RMS_EXT_OVER_B	RMS Extreme Over Voltage Period PhB (optional)
ZCL_ELEC_MEAS_ATTR_RMS_EXT_UNDER_B	RMS Extreme Under Voltage Period PhB (optional)
ZCL_ELEC_MEAS_ATTR_RMS_SAG_PERIOD_B	RMS Voltage Sag Period PhB (optional)
ZCL_ELEC_MEAS_ATTR_RMS_SWELL_PERIOD_B	RMS Voltage Swell Period PhB (optional)
ZCL_ELEC_MEAS_ATTR_LINE_CURR_C	Line Current PhC (optional)
ZCL_ELEC_MEAS_ATTR_ACTIVE_CURR_C	Active Current PhC (optional)
ZCL_ELEC_MEAS_ATTR_REACTIVE_CURR_C	Reactive Current PhC (optional)
ZCL_ELEC_MEAS_ATTR_RMS_VOLT_C	RMS Voltage PhC (optional)
ZCL_ELEC_MEAS_ATTR_RMS_VOLT_MIN_C	RMS Voltage Min PhC (optional)
ZCL_ELEC_MEAS_ATTR_RMS_VOLT_MAX_C	RMS Voltage Max PhC (optional)
ZCL_ELEC_MEAS_ATTR_RMS_CURR_C	RMS Current PhC (optional)
ZCL_ELEC_MEAS_ATTR_RMS_CURR_MIN_C	RMS Current Min PhC (optional)
ZCL_ELEC_MEAS_ATTR_RMS_CURR_MAX_C	RMS Current Max PhC (optional)
ZCL_ELEC_MEAS_ATTR_ACTIVE_PWR_C	Active Power PhC (optional)
ZCL_ELEC_MEAS_ATTR_ACTIVE_PWR_MIN_C	Active Power Min PhC (optional)
ZCL_ELEC_MEAS_ATTR_ACTIVE_PWR_MAX_C	Active Power Max PhC (optional)
ZCL_ELEC_MEAS_ATTR_REACTIVE_PWR_C	Reactive Power PhC (optional)
ZCL_ELEC_MEAS_ATTR_APPARENT_PWR_C	Apparent Power PhC (optional)
ZCL_ELEC_MEAS_ATTR_PWR_FACTOR_C	Power Factor PhC (optional)
ZCL_ELEC_MEAS_ATTR_AVG_RMS_VOLT_PERIOD_ C	Average RMS Voltage Measurement Period PhC (optional)
ZCL_ELEC_MEAS_ATTR_AVG_RMS_OV_C	Average RMS Over Voltage Counter PhC (optional)
ZCL_ELEC_MEAS_ATTR_AVG_RMS_UV_C	Average RMS Under Voltage Counter PhC (optional)
ZCL_ELEC_MEAS_ATTR_RMS_EXT_OVER_C	RMS Extreme Over Voltage Period PhC (optional)
ZCL_ELEC_MEAS_ATTR_RMS_EXT_UNDER_C	RMS Extreme Under Voltage Period PhC (optional)
ZCL_ELEC_MEAS_ATTR_RMS_SAG_PERIOD_C	RMS Voltage Sag Period PhC (optional)
ZCL_ELEC_MEAS_ATTR_RMS_SWELL_PERIOD_C	RMS Voltage Swell Period PhC (optional)

5.12.4 Electrical measurement cluster structures

5.12.4.1 *ZbZclElecMeasClientGetMeasProfileReqT*

Get Measurement Profile command structure

Table 237. **ZbZclElecMeasClientGetMeasProfileReqT parameters**

Parameter	Description
`uint16_t attr_id`	`uint16_t attr_id`
`uint32_t start_time`	Start Time
`uint8_t num_intervals`	Number of intervals

5.12.4.2 *ZbZclElecMeasSvrCallbacksT*

Electrical Measurement Server callbacks configuration

Table 238. ZbZclElecMeasSvrCallbacksT Parameters

Paramter	Descption
`get_profile_info` (callback function pointer)	`Enum ZclStatusCodeT (*get_profile_info)(struct ZbZclClusterT *clusterPtr, struct ZbZclAddrInfoT*src_info, void *arg)` Callback to application, invoked on receipt of Get Profile Info command.
`get_meas_profile` (callback function pointer)	`Enum ZclStatusCodeT (*get_meas_profile)(struct ZbZclClusterT *clusterPtr, struct ZbZclElecMeasClientGetMeasProfileReqT*cmd_req,struct ZbZclAddrInfoT *src_info, void *arg)` Callback to application, invoked on receipt of Get Measurement Profile command.

5.12.4.3 *ZbZclElecMeasSvrGetMeasProfileRspT*

Get Measurement Profile response structure

Table 239. ZbZclElecMeasSvrGetMeasProfileRspT parameters

Parameter	Description
`uint32_t start_time`	Start time
`uint8_t status`	Status
`uint8_t profile_interval_period`	Profile interval period
`uint8_t num_intervals_delivered`	Number Of Intervals Delivered
`uint16_t attr_id`	Attribute ID
`uint8_t *interval_data`	Intervals
`uint16_t interval_len`	Number of Intervals

5.12.4.4 *ZbZclElecMeasSvrGetProfileInfoRspT*

Get Profile Info response structure

Table 240. ZbZclElecMeasSvrGetProfileInfoRspT parameters

Parameter	Description
`uint8_t profile_count`	Profile Count
`uint8_t profile_interval_period`	Profile interval period
`uint8_t max_num_intervals`	Max number of intervals
`uint16_t *attr_list`	List of attributes

5.13 Fan control cluster

```
#include "zcl/general/zcl.fan.h"
```

5.13.1 Fan control cluster description

Table 241. Fan control cluster PICS code description

PICS Code	Name	Supported	Notes
FAN.S	Server	True	-
FAN.C	Client	True	-
FAN.S.A0000	`FanMode` server attribute	True	-
FAN.S.A0001	`FanModeSequence` server attribute	True	-
FAN.S.Afffd	`ClusterRevision` server attribute	True	-
FAN.S.Afffe	`AttributeReportingStatus` server attribute	False	-
FAN.C.Afffd	`ClusterRevision` client attribute	True	-
FAN.C.Afffe	`AttributeReportingStatus` client attribute	False	-

5.13.2 Fan control cluster functions

5.13.2.1 ZbZclFanClientAlloc

```
struct ZbZclClusterT * ZbZclFanClientAlloc(struct ZigBeeT *zb, uint8_t endpoint);
```

Create a new instance of the Fan Control Client cluster

Table 242. ZbZclFanClientAlloc Parameters

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster

Return: Cluster pointer, or NULL if there is an error.

5.13.2.2 ZbZclFanServerAlloc

```
struct ZbZclClusterT * ZbZclFanServerAlloc(struct ZigBeeT *zb, uint8_t endpoint);
```

Create a new instance of the Fan Control Server cluster

Table 243. ZbZclFanServerAlloc parameters

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster

Return: Cluster pointer, or NULL if there is an error.

5.13.3 Fan control cluster enumerations

5.13.3.1 ZbZclFanModeT

Fan Mode Attribute Values

Table 244. **ZbZclFanModeT attributes**

Attribute	Description
ZCL_FAN_MODE_OFF	Off
ZCL_FAN_MODE_LOW	Low
ZCL_FAN_MODE_MED	Medium
ZCL_FAN_MODE_HI	High
ZCL_FAN_MODE_ON	On
ZCL_FAN_MODE_AUTO	Auto (the fan speed is self-regulated)
ZCL_FAN_MODE_SMART	Smart (when the heated/cooled space is occupied, the fan is always on)

5.13.3.2 *ZbZclFanSeqT*

Fan Sequence Operation Attribute Values

Table 245. **ZbZclFanSeqT attributes**

Attribute	Description
ZCL_FAN_SEQ_LMH	Low/Med/High
ZCL_FAN_SEQ_LH	Low/High
ZCL_FAN_SEQ_LMHA	Low/Med/High/Auto
ZCL_FAN_SEQ_LHA	Low/High/Auto
ZCL_FAN_SEQ_OA	On/Auto

5.13.3.3 *ZbZclFanSvrAttrT*

Fan Control server attribute IDs

Table 246. **ZbZclFanSvrAttrT attributes**

Attribute	Description
ZCL_FAN_ATTR_MODE	Fan Mode
ZCL_FAN_ATTR_SEQUENCE	Fan Mode Sequence

5.14 Groups cluster

```
#include "zcl/general/zcl.groups.h"
```

5.14.1 Groups cluster description

Table 247. **Groups cluster PICS code description**

PICS Code	Name	Supported	Notes
G.S	Server	True	-
G.C	Client	True	-
G.S.A0000	`NameSupport` server attribute	True	-
G.S.Afffd	`ClusterRevision` server attribute	True	-
G.S.C00.Rsp	Add group server command	True	-
G.S.C01.Rsp	View group server command	True	-
G.S.C02.Rsp	Get group membership server command	True	-

PICS Code	Name	Supported	Notes
G.S.C03.Rsp	Remove group server command	True	-
G.S.C04.Rsp	Remove all groups server command	True	-
G.S.C05.Rsp	Add group if identifying server command	True	-
G.S.C00.Tx	Add group response server command	True	-
G.S.C01.Tx	View group response server command	True	-
G.S.C02.Tx	Get group membership response server command	True	-
G.S.C03.Tx	Remove group response server command	True	-
G.C.Afffd	`ClusterRevision` client attribute	True	-
G.C.C00.Rsp	Add group response client command	True	-
G.C.C01.Rsp	View group response client command	True	-
G.C.C06.Rsp	Get group membership response client command	True	-
G.C.C02.Rsp	Remove group response client command	True	-
G.C.C00.Tx	Add group client command	True	-
G.C.C01.Tx	View group client command	True	-
G.C.C02.Tx	Get group membership client command	True	-
G.C.C03.Tx	Remove group client command	True	-
G.C.C04.Tx	Remove all groups client command	True	-
G.C.C05.Tx	Add group if identifying client command	True	-

5.14.2 Groups cluster functions

5.14.2.1 *ZbZclGroupsClientAddIdentifyingReq*

```
enum ZclStatusCodeT ZbZclGroupsClientAddIdentifyingReq(struct ZbZclClusterT*cluster, struct
ZbZclGroupsClientAddIdentifyingReqT *req, void (*callback)(struct ZbZclCommandRspT *rsp,
void *arg), void *arg);
```

Send an Add Group If Identifying command

Table 248. **ZbZclGroupsClientAddIdentifyingReq Parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`req`	Add Group If Identifying command request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.14.2.2 *ZbZclGroupsClientAddReq*

```
enum ZclStatusCodeT ZbZclGroupsClientAddReq(struct ZbZclClusterT *cluster, struct
ZbZclGroupsClientAddReqT *req, void (*callback)(struct ZbZclCommandRspT *rsp, void*arg),
void *arg);
```

Send an Add Group command

Table 249. ZbZclGroupsClientAddReq parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`req`	Add Group command request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.14.2.3 *ZbZclGroupsClientAlloc*

```
struct ZbZclClusterT * ZbZclGroupsClientAlloc(struct ZigBeeT *zb, uint8_t
```

Create a new instance of the Groups Client cluster

Table 250. ZbZclGroupsClientAlloc parameters

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster

Return: Cluster pointer, or NULL if there is an error.

5.14.2.4 *ZbZclGroupsClientGetMembershipReq*

```
enum ZclStatusCodeT ZbZclGroupsClientGetMembershipReq(struct ZbZclClusterT*cluster, struct
ZbZclGroupsClientGetMembershipReqT *req, void (*callback)(struct ZbZclCommandRspT *rsp, void
*arg), void *arg);
```

Send a Get Group Membership command

Table 251. ZbZclGroupsClientGetMembershipReq parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`req`	Get Group Membership command request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.14.2.5 *ZbZclGroupsClientRemoveAllReq*

```
enum ZclStatusCodeT ZbZclGroupsClientRemoveAllReq(struct ZbZclClusterT *cluster, struct
ZbApsAddrT *dst, void (*callback)(struct ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Remove All Groups command

Table 252. ZbZclGroupsClientRemoveAllReq parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`req`	Remove All Group command request structure

Parameter	Description
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.14.2.6 *ZbZclGroupsClientRemoveReq*

```
enum ZclStatusCodeT ZbZclGroupsClientRemoveReq(struct ZbZclClusterT *cluster, struct
ZbZclGroupsClientRemoveReqT *req, void (*callback)(struct ZbZclCommandRspT*rsp, void *arg),
void *arg);
```

Send a Remove Group command

Table 253. **ZbZclGroupsClientRemoveReq parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`req`	Remove Group command request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.14.2.7 *ZbZclGroupsClientViewReq*

```
enum ZclStatusCodeT ZbZclGroupsClientViewReq(struct ZbZclClusterT *cluster, struct
ZbZclGroupsClientViewReqT *req, void (*callback)(struct ZbZclCommandRspT *rsp, void*arg),
void *arg);
```

Send a View Group command

Table 254. **ZbZclGroupsClientViewReq parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`req`	View Group command request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.14.2.8 *ZbZclGroupsServerAlloc*

```
struct ZbZclClusterT * ZbZclGroupsServerAlloc(struct ZigBeeT *zb, uint8_t endpoint);
```

Create a new instance of the Groups Server cluster

Table 255. **ZbZclGroupsServerAlloc Parameters**

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster

Return: Cluster pointer, or NULL if there is an error.

5.14.3 Groups cluster enumerations

5.14.3.1 *ZbZclGroupsSvrAttrT*

Table 256. Groups server attribute IDs

Attributes	Description
ZCL_GROUPS_ATTR_NAME_SUPPORT	Name Support

5.14.4 Groups cluster structures

5.14.4.1 *ZbZclGroupsClientAddIdentifyingReqT*

Add Group If Identifying command structure

Table 257. ZbZclGroupsClientAddIdentifyingReqT parameters

Parameter	Description
`struct ZbApsAddrT dst`	Destination Address
`uint16_t group_id`	Group ID

5.14.4.2 *ZbZclGroupsClientAddReqT*

Add Group command structure

Table 258. ZbZclGroupsClientAddReqT parameters

Parameter	Description
`struct ZbApsAddrT dst`	Destination Address
`uint16_t group_id`	Group ID

5.14.4.3 *ZbZclGroupsClientGetMembershipReqT*

Get Group Membership command structure

Table 259. ZbZclGroupsClientGetMembershipReqT parameters

Parameter	Description
`struct ZbApsAddrT dst`	Destination Address
`uint8_t num_groups`	Group count
`uint16_t group_list`	Group list

5.14.4.4 *ZbZclGroupsClientRemoveReqT*

Remove Group command structure

Table 260. ZbZclGroupsClientRemoveReqT parameters

Parameter	Description
`struct ZbApsAddrT dst`	Destination Address
`uint16_t group_id`	Group ID

5.14.4.5 *ZbZclGroupsClientViewReqT*

View Group command structure

Table 261. ZbZclGroupsClientViewReqT parameters

Parameter	Description
`struct ZbApsAddrT dst`	Destination Address
`uint16_t group_id`	Group ID

5.15 IAS ACE cluster

```
#include "zcl/security/zcl.ias_ace.h"
```

5.15.1 IAS ACE cluster description

Table 262. IAS ACE cluster PICS code description

PICS Code	Name	Supported	Notes
IASACE.S	Server	True	-
IASACE.C	Client	True	-
IASACE.S.Afffd	`ClusterRevision` server attribute	True	-
IASACE.S.Affe	`AttributeReportingStatus` server attribute	False	-
IASACE.S.C00.Rsp	Arm server command	True	-
IASACE.S.C01.Rsp	Bypass server command	True	-
IASACE.S.C02.Rsp	Emergency server command	True	-
IASACE.S.C03.Rsp	Fire server command	True	-
IASACE.S.C04.Rsp	Panic server command	True	-
IASACE.S.C05.Rsp	Get Zone ID Map server command	True	-
IASACE.S.C06.Rsp	Get Zone Information server command	True	-
IASACE.S.C07.Rsp	Get Panel Status server command	True	-
IASACE.S.C08.Rsp	Get Bypassed Zone List server command	True	-
IASACE.S.C09.Rsp	Get Zone Status server command	True	-
IASACE.S.C00.Tx	Arm Response server command	True	-
IASACE.S.C01.Tx	Get Zone ID Map Response server command	True	-
IASACE.S.C02.Tx	Get Zone Information Response server command	True	-
IASACE.S.C03.Tx	Zone Status Changed server command	True	-
IASACE.S.C04.Tx	Panel Status Changed server command	True	-
IASACE.S.C05.Tx	Get Panel Status Response server command	True	-
IASACE.S.C06.Tx	Set Bypassed Zone List server command	True	-
IASACE.S.C07.Tx	Bypass Response server command	True	-
IASACE.S.C08.Tx	Get Zone Status Response server command	True	-
IASACE.C.Afffd	`ClusterRevision` client attribute	True	-
IASACE.C.Affe	`AttributeReportingStatus` client attribute	False	-
IASACE.C.C00.Tx	Arm client command	True	-
IASACE.C.C01.Tx	Bypass client command	True	-

PICS Code	Name	Supported	Notes
IASACE.C.C02.Tx	Emergency client command	True	-
IASACE.C.C03.Tx	Fire client command	True	-
IASACE.C.C04.Tx	Panic client command	True	-
IASACE.C.C05.Tx	Get Zone ID Map client command	True	-
IASACE.C.C06.Tx	Get Zone Information client command	True	-
IASACE.C.C07.Tx	Get Panel Status client command	True	-
IASACE.C.C08.Tx	Get Bypassed Zone List client command	True	-
IASACE.C.C09.Tx	Get Zone Status client command	True	-
IASACE.C.C00.Rsp	Arm Response client command	True	-
IASACE.C.C01.Rsp	Get Zone ID Map Response client command	True	-
IASACE.C.C02.Rsp	Get Zone Information Response client command	True	-
IASACE.C.C03.Rsp	Zone Status Changed client command	True	-
IASACE.C.C04.Rsp	Panel Status Changed client command	True	-
IASACE.C.C05.Rsp	Get Panel Status Response client command	True	-
IASACE.C.C06.Rsp	Set Bypassed Zone List client command	True	-
IASACE.C.C07.Rsp	Bypass Response client command	True	-
IASACE.C.C08.Rsp	Get Zone Status Response client command	True	-

5.15.2 IAS ACE cluster functions

5.15.2.1 *ZbZclIasAceClientAlloc*

```
struct ZbZclClusterT * ZbZclIasAceClientAlloc(struct ZigBeeT *zb, uint8_t endpoint, void
*arg);
```

Allocate the IAS ACE Client cluster

Table 263. **ZbZclIasAceClientAlloc parameters**

Parameter	Description
zb	Zigbee® stack instance
endpoint	Endpoint on which to create cluster
arg	Pointer to application data that is provided back to the callback function when invoked

Return: Cluster pointer, or NULL if there is an error.

5.15.2.2 *ZbZclIasAceClientCommandArmReq*

```
uint8_t ZbZclIasAceClientCommandArmReq(struct ZbZclClusterT *cluster, const struct
ZbApsAddrT *dst, struct ZbZclIasAceClientCommandArmT *cmd_req, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send an Arm command

Table 264. **ZbZclIasAceClientCommandArmReq parameters**

Parameter	Description
cluster	Cluster instance from which to send this command
dst	Requested destination address

Parameter	Description
`cmd_req`	Arm® command request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.15.2.3 *ZbZclIasAceClientCommandBypassReq*

```
uint8_t ZbZclIasAceClientCommandBypassReq(struct ZbZclClusterT *cluster, const struct
ZbApsAddrT *dst, struct ZbZclIasAceClientCommandBypassT *cmd_req, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Bypass command

Table 265. **ZbZclIasAceClientCommandBypassReq parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`cmd_req`	Bypass command request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.15.2.4 *ZbZclIasAceClientCommandEmergencyReq*

```
uint8_t ZbZclIasAceClientCommandEmergencyReq(struct ZbZclClusterT *cluster, const struct
ZbApsAddrT *dst, void (*callback)(struct ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send an Emergency command

Table 266. **ZbZclIasAceClientCommandEmergencyReq parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.15.2.5 *ZbZclIasAceClientCommandFireReq*

```
uint8_t ZbZclIasAceClientCommandFireReq(struct ZbZclClusterT *cluster, const struct
ZbApsAddrT *dst, void (*callback)(struct ZbZclCommandRspT *rsp, void *arg), void*arg);
```

Send a Fire command

Table 267. ZbZclIasAceClientCommandFireReq parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.15.2.6 *ZbZclIasAceClientCommandGetBypassedZoneListReq*

```
uint8_t ZbZclIasAceClientCommandGetBypassedZoneListReq(struct ZbZclClusterT*cluster, const
struct ZbApsAddrT *dst, void (*callback)(struct ZbZclCommandRspT*rsp, void *arg), void *arg);
```

Send a Get Bypassed Zone List command

Table 268. ZbZclIasAceClientCommandGetBypassedZoneListReq parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.15.2.7 *ZbZclIasAceClientCommandGetPanelStatusReq*

```
uint8_t ZbZclIasAceClientCommandGetPanelStatusReq(struct ZbZclClusterT *cluster, const
struct ZbApsAddrT *dst, void (*callback)(struct ZbZclCommandRspT *rsp, void*arg), void *arg);
```

Send a Get Panel Status changed request

Table 269. ZbZclIasAceClientCommandGetPanelStatusReq parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.15.2.8 *ZbZclIasAceClientCommandGetZoneIdMapReq*

```
uint8_t ZbZclIasAceClientCommandGetZoneIdMapReq(struct ZbZclClusterT *cluster, const struct
ZbApsAddrT *dst, void (*callback)(struct ZbZclCommandRspT *rsp, void*arg), void *arg);
```

Send a Get Zone ID Map command

Table 270. ZbZclIasAceClientCommandGetZoneIdMapReq parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.15.2.9 *ZbZclIasAceClientCommandGetZoneInfoReq*

```
uint8_t ZbZclIasAceClientCommandGetZoneInfoReq(struct ZbZclClusterT *cluster, const struct
ZbApsAddrT *dst, struct ZbZclIasAceClientCommandGetZoneInfoT *cmd_req, void (*callback)
(struct ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Get Zone Info command

Table 271. ZbZclIasAceClientCommandGetZoneInfoReq parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`cmd_req`	Get Zone Info command request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.15.2.10 *ZbZclIasAceClientCommandGetZoneStatusReq*

```
uint8_t ZbZclIasAceClientCommandGetZoneStatusReq(struct ZbZclClusterT *cluster, const struct
ZbApsAddrT *dst, struct ZbZclIasAceClientCommandGetZoneStatusT*cmd_req, void (*callback)
(struct ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Get Zone Status command

Table 272. ZbZclIasAceClientCommandGetZoneStatusReq parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`cmd_req`	Get Zone Status command request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.15.2.11 *ZbZclIasAceClientCommandPanicReq*

```
uint8_t ZbZclIasAceClientCommandPanicReq(struct ZbZclClusterT *cluster, const struct
ZbApsAddrT *dst, void (*callback)(struct ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Panic command

Table 273. ZbZclIasAceClientCommandPanicReq parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.15.2.12 *ZbZclIasAceClientParseArmRsp*

```
bool ZbZclIasAceClientParseArmRsp(const uint8_t *buf, unsigned int len, struct
ZbZclIasAceServerCommandArmRspT *rsp);
```

Parse an Arm Response command payload into a data structure

Table 274. ZbZclIasAceClientParseArmRsp parameters

Parameter	Description
`buf`	Response buffer
`len`	Length of response buffer
`rsp`	Arm® response structure

Return: True on success, false otherwise

5.15.2.13 *ZbZclIasAceClientParseBypassRsp*

```
bool ZbZclIasAceClientParseBypassRsp(const uint8_t *buf, unsigned int len, struct
ZbZclIasAceServerCommandBypassRspT *rsp);
```

Parse a Bypass Response command payload into a data structure

Table 275. ZbZclIasAceClientParseBypassRsp parameters

Parameter	Description
`buf`	Response buffer
`len`	Length of response buffer
`rsp`	Bypass Response structure

Return: True on success, false otherwise

5.15.2.14 *ZbZclIasAceClientParseGetPanelStatusRsp*

```
bool ZbZclIasAceClientParseGetPanelStatusRsp(const uint8_t *buf, unsigned int len, struct
ZbZclIasAceServerCommandGetPanelStatusRspT *rsp);
```

Parse a Get Panel Status Response command payload into a data structure

Table 276. ZbZclIasAceClientParseGetPanelStatusRsp parameters

Parameter	Description
`buf`	Response buffer
`len`	Length of response buffer

Parameter	Description
rsp	Zone Status Changed Response structure

Return: True on success, false otherwise

5.15.2.15 *ZbZclIasAceClientParseGetZoneIdMapRsp*

```
bool ZbZclIasAceClientParseGetZoneIdMapRsp(const uint8_t *buf, unsigned int len, struct
ZbZclIasAceServerCommandGetZoneIdMapRspT *rsp);
```

Parse a Get Zone ID Map Response command payload into a data structure

Table 277. ZbZclIasAceClientParseGetZoneIdMapRsp parameters

Parameter	Description
buf	Response buffer
len	Length of response buffer
rsp	Get Zone ID Map Response structure

Return: True on success, false otherwise

5.15.2.16 *ZbZclIasAceClientParseGetZoneInfoRsp*

```
bool ZbZclIasAceClientParseGetZoneInfoRsp(const uint8_t *buf, unsigned int len, struct
ZbZclIasAceServerCommandGetZoneInfoRspT *rsp);
```

Parse a Get Zone Info Response command payload into a data structure

Table 278. ZbZclIasAceClientParseGetZoneInfoRsp parameters

Parameter	Description
buf	Response buffer
len	Length of response buffer
rsp	Get Zone Info Response structure

Return: True on success, false otherwise

5.15.2.17 *ZbZclIasAceClientParseGetZoneStatusRsp*

```
bool ZbZclIasAceClientParseGetZoneStatusRsp(const uint8_t *buf, unsigned int len, struct
ZbZclIasAceServerCommandGetZoneStatusRspT *rsp);
```

Parse a Get Zone Status Response command payload into a data structure

Table 279. ZbZclIasAceClientParseGetZoneStatusRsp parameters

Parameter	Description
buf	Response buffer
len	Length of response buffer
rsp	Get Zone Status Response structure

Return: True on success, false otherwise

5.15.2.18 *ZbZclIasAceClientParseSetBypassedZoneList*

```
bool ZbZclIasAceClientParseSetBypassedZoneList(const uint8_t *buf, unsigned int len, struct
ZbZclIasAceServerCommandSetBypassedZoneListT *rsp);
```

Parse a Set Bypassed Zone List Response command payload into a data structure

Table 280. ZbZclIasAceClientParseSetBypassedZoneList parameters

Parameter	Description
`buf`	Response buffer
`len`	Length of response buffer
`rsp`	Set Bypassed Zone List Response structure

Return: True on success, false otherwise

5.15.2.19 *ZbZclIasAceClientParseZoneStatusChanged*

```
bool ZbZclIasAceClientParseZoneStatusChanged(const uint8_t *buf, unsigned int len, struct
ZbZclIasAceServerCommandZoneStatusChangedT *rsp);
```

Parse a Zone Status Changed Response command payload into a data structure

Table 281. ZbZclIasAceClientParseZoneStatusChanged parameters

Parameter	Description
`buf`	Response buffer
`len`	Length of response buffer
`rsp`	Zone Status Changed Response structure

Return: True on success, false otherwise

5.15.2.20 *ZbZclIasAceServerAlloc*

```
struct ZbZclClusterT * ZbZclIasAceServerAlloc(struct ZigBeeT *zb, uint8_t endpoint, struct
ZbZclIasAceServerCallbacksT *callbacks, void *arg);
```

Create a new instance of the IAS ACE Server cluster

If 'use_trip_pair' is true, application must call `ZbZclIasAceServerEnrollRequest` to perform the 'trip-to-pair' process, unless the IAS CIE has sent us an unsolicited Auto-Enroll-Response

Table 282. ZbZclIasAceServerAlloc parameters

Paramter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster
`callbacks`	Structure containing any callback function pointers for this cluster
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: Cluster pointer, or NULL if there is an error.

5.15.2.21 *ZbZclIasAceServerGetFreeZoneId*

```
bool ZbZclIasAceServerGetFreeZoneId(struct ZbZclClusterT *cluster, uint8_t*zone_id_ptr);
```

Returns the first free Zone ID not already in the Zone Table

Table 283. ZbZclIasAceServerGetFreeZoneId parameters

Parameter	Descripiton
`cluster`	Cluster instance from which to send this command
`zone_id_ptr`	Zone ID value

Return: True on success, false otherwise

5.15.2.22 *ZbZclIasAceServerPanelCodeConfig*

```
bool ZbZclIasAceServerPanelCodeConfig(struct ZbZclClusterT *cluster, const char*arm_code);
```

Change the Panel Arm/Disarm Code

Table 284. ZbZclIasAceServerPanelCodeConfig parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`arm_code`	Arm® code

Return: True on success, false otherwise

5.15.2.23 *ZbZclIasAceServerPanelStatusConfig*

```
bool ZbZclIasAceServerPanelStatusConfig(struct ZbZclClusterT *cluster, enum
ZbZclIasAcePanelStatusT panel_status, uint8_t seconds_remain, enum ZbZclIasAceAudibleNotifyT
audible_notify);
```

Update the Panel Status

Table 285. ZbZclIasAceServerPanelStatusConfig parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`panel_status`	Panel Status enumeration
`seconds_remain`	Seconds remaining

Return: True on success, false otherwise

5.15.2.24 *ZbZclIasAceServerZoneBypassConfig*

```
enum ZbZclIasAceBypassResultT ZbZclIasAceServerZoneBypassConfig(struct ZbZclClusterT
*cluster, uint8_t zone_id, bool bypass);
```

Bypass zone if allowed

Table 286. ZbZclIasAceServerZoneBypassConfig parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`zone_id`	Zone ID of zone in question
`bypass`	True if zone is bypassed, false if not

Return: Result of bypass config command

5.15.2.25 *ZbZclIasAceServerZoneBypassPerms*

```
bool ZbZclIasAceServerZoneBypassPerms(struct ZbZclClusterT *cluster, uint8_t zone_id, enum
ZbZclIasAceBypassPermsT bypass_perms);
```

Configure Bypass Permissions

Table 287. ZbZclIasAceServerZoneBypassPerms parameters

Parameter	Description
cluster	Cluster instance from which to send this command
zone_id	Zone ID of zone in question
bypass_perms	Desired bypass permissions

Return: True on success, false otherwise

5.15.2.26 *ZbZclIasAceServerZoneStatusConfig*

```
bool ZbZclIasAceServerZoneStatusConfig(struct ZbZclClusterT *cluster, uint8_t zone_id, enum
ZbZclIasZoneServerZoneStatusT zone_status, enum ZbZclIasAceAudibleNotifyT audible_notify);
```

Configure Zone Status

Table 288. ZbZclIasAceServerZoneStatusConfig parameters

Parameter	Description
cluster	Cluster instance from which to send this command
zone_id	Zone ID of zone in question
zone_status	Desired zone status
audible_notify	Determines if zone notification is audible or not

Return: True on success, false otherwise

5.15.2.27 *ZbZclIasAceServerZoneTableAdd*

```
bool ZbZclIasAceServerZoneTableAdd(struct ZbZclClusterT *cluster, struct
ZbZclIasAceServerZoneTableAddT *req);
```

Add new zone entry

Table 289. ZbZclIasAceServerZoneTableAdd parameters

Parameter	Description
cluster	Cluster instance from which to send this command
req	Zone Table Add request structure

Return: True on success, false otherwise

5.15.2.28 *ZbZclIasAceServerZoneTableAddrLookup*

```
uint64_t ZbZclIasAceServerZoneTableAddrLookup(struct ZbZclClusterT *cluster, uint8_t
zone_id);
```

Returns address of paired zone, or 0 if not found

Table 290. ZbZclIasAceServerZoneTableAddrLookup parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`zone_id`	ID of zone to be returned

Return: Address of zone if successful, 0 on error

5.15.2.29 *ZbZclIasAceServerZoneTableDeleteByAddr*

```
bool ZbZclIasAceServerZoneTableDeleteByAddr(struct ZbZclClusterT *cluster, uint64_t addr);
```

Delete a zone by address

Table 291. ZbZclIasAceServerZoneTableDeleteByAddr parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`addr`	Address of zone to be deleted

Return: True on success, false otherwise

5.15.2.30 *ZbZclIasAceServerZoneTableDeleteById*

```
bool ZbZclIasAceServerZoneTableDeleteById(struct ZbZclClusterT *cluster, uint8_t zone_id);
```

Delete a zone by zone ID

Table 292. ZbZclIasAceServerZoneTableDeleteById parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`zone_id`	ID of zone to be deleted

Return: True on success, false otherwise

5.15.2.31 *ZbZclIasAceServerZoneTableIdLookup*

```
bool ZbZclIasAceServerZoneTableIdLookup(struct ZbZclClusterT *cluster, uint64_t zone_addr,
uint8_t *zone_id_ptr);
```

Attempts to find a zone based on extended address, and returns the zone Id if found

Table 293. ZbZclIasAceServerZoneTableIdLookup parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`zone_addr`	Address of Zone being looked up
`zone_id_ptr`	If successful, points to zone ID that was looked up

Return: True on success, false otherwise

5.15.3 IAS ACE cluster structures

5.15.3.1 *ZbZclIasAceClientCommandArmT*

Arm command structure

Table 294. ZbZclIasAceClientCommandArmT parameters

Parameter	Description
`enum ZbZclIasAceArmModeTarm_mode`	Arm® Mode
`char arm_code`	Arm/Disarm Code
`uint8_t zone_id`	Zone ID

5.15.3.2 *ZbZclIasAceClientCommandBypassT*

Bypass command structure

Table 295. ZbZclIasAceClientCommandBypassT parameters

Parameter	Description
`uint8_t num_zones`	Number of Zones
`uint8_t zone_id_list`	Zone ID List
`char arm_code`	Arm/Disarm Code

5.15.3.3 *ZbZclIasAceClientCommandGetZoneInfoT*

Get Zone Info command structure

Table 296. ZbZclIasAceClientCommandGetZoneInfoT parameters

Parameter	Description
`uint8_t zone_id`	Zone ID

5.15.3.4 *ZbZclIasAceClientCommandGetZoneStatusT*

Get Zone Status command structure

Table 297. ZbZclIasAceClientCommandGetZoneStatusT parameters

Parameter	Description
`uint8_t starting_zone_id`	Starting Zone ID
`uint8_t max_zone_ids`	Max Number of Zone IDs
`uint8_t zone_status_mask_flag`	Zone Status Mask Flag
`uint16_t zone_status_mask`	Zone Status Mask

5.15.3.5 *ZbZclIasAceServerCallbacksT*

IAS ACE Server callbacks configuration

Table 298. ZbZclIasAceServerCallbacksT parameters

Parameter	Description
arm_req (callback function pointer)	`bool (*arm_req)(struct ZbZclClusterT *clusterPtr, void*arg,struct ZbZclIasAceClientCommandArmT *arm_req,struct ZbZclIasAceServerCommandArmRspT *arm_rsp)` Callback to application, invoked on receipt of Arm command

Parameter	Description
bypass_req (callback function pointer)	`Void (*bypass_req)(structZbZclClusterT *clusterPtr, void *arg,struct ZbZclIasAceClientCommandBypassT*bypass_req,struct ZbZclIasAceServerCommandBypassRspT*bypass_rsp)` Callback to application, invoked on receipt of Bypass command
emerg_req (callback function pointer)	`uint8_t (*emerg_req)(struct ZbZclClusterT *clusterPtr, void *arg, struct ZbZclAddrInfoT *srcInfo)` Callback to application, invoked on receipt of Emergency command
fire_req (callback function pointer)	`uint8_t (*fire_req)(struct ZbZclClusterT *clusterPtr, void *arg, struct ZbZclAddrInfoT *srcInfo)` Callback to application, invoked on receipt of Fire command
panic_req (callback function pointer)	`uint8_t (*panic_req)(struct ZbZclClusterT *clusterPtr, void *arg, struct ZbZclAddrInfoT *srcInfo)` Callback to application, invoked on receipt of Panic command

5.15.3.6 *ZbZclIasAceServerCommandArmRspT*

Arm response structure

Table 299. ZbZclIasAceServerCommandArmRspT parameters

Parameter	Description
`enum ZbZclIasAceArmNotifyTarm_notify`	Arm® Notification

5.15.3.7 *ZbZclIasAceServerCommandBypassRspT*

Bypass Response response structure

Table 300. bZclIasAceServerCommandBypassRspTparameters

Parameter	Description
`uint8_t num_zones`	Number of Zones
`enum ZbZclIasAceBypassResultTbypass_result_list`	Bypass Result for Zone ID List

5.15.3.8 *ZbZclIasAceServerCommandGetPanelStatusRspT*

Get Panel Status response structure

Table 301. ZbZclIasAceServerCommandGetPanelStatusRspT parameters

Parameter	Description
`enum ZbZclIasAcePanelStatusTpanel_status`	Panel Status
`uint8_t seconds_remain`	Seconds Remaining
`enum ZbZclIasAceAudibleNotifyTaudible_notify`	Audible Notification
`enum ZbZclIasAceAlarmStatusTalarm_status`	Alarm Status

5.15.3.9 *ZbZclIasAceServerCommandGetZoneIdMapRspT*

Get Zone ID Map response structure

Table 302. ZbZclIasAceServerCommandGetZoneIdMapRspT parameters

Parameter	Description
`uint16_t zond_id_map_list`	Zone ID Map List

5.15.3.10 *ZbZclIasAceServerCommandGetZoneInfoRspT*

Get Zone Info response structure

Table 303. ZbZclIasAceServerCommandGetZoneInfoRspT parameters

Parameter	Description
`uint8_t zone_id`	Zone ID
`enum ZbZclIasZoneServerZoneTypeT zone_type`	Zone Type
`uint64_t zone_addr`	IEEE Address
`char zone_label`	Zone Label

5.15.3.11 *ZbZclIasAceServerCommandGetZoneStatusRspT*

Get Zone Status response structure

Table 304. ZbZclIasAceServerCommandGetZoneStatusRspT parameters

Parameter	Description
`uint8_t zone_status_complete`	Zone Status Complete
`uint8_t num_zones`	Number of Zones
`struct {uint8_t zone_id`	Zone ID
`enum ZbZclIasZoneServerZoneStatusT zone_status`	Zone Status

5.15.3.12 *ZbZclIasAceServerCommandSetBypassedZoneListT*

Set Bypassed Zone List command structure

Table 305. ZbZclIasAceServerCommandSetBypassedZoneListT parameters

Parameter	Description
`uint8_t num_zones`	Number of Zones
`uint8_t zone_id_list`	Zone ID List

5.15.3.13 *ZbZclIasAceServerCommandZoneStatusChangedT*

Zone Status Changed command structure

Table 306. ZbZclIasAceServerCommandZoneStatusChangedT parameters

Parameter	Description
`uint8_t zone_id`	Zone ID
`enum ZbZclIasZoneServerZoneStatusT zone_status`	Zone Status
`enum ZbZclIasAceAudibleNotifyTaudible_notify`	Audible Notification
`char zone_label`	Zone Label

5.15.3.14 *ZbZclIasAceServerZoneTableAddT*

Zone Table Add request structure

Table 307. ZbZclIasAceServerZoneTableAddT parameters

Parameter	Description
`enum ZbZclIasZoneServerZoneTypeT zone_type`	Zone Type
`uint64_t zone_addr`	Zone Address
`const char *zone_label`	Zone Label – may be Max length = NULL, ZCL_IAS_ACE_ZONE_LABEL_STRING_MAX_LEN
`uint8_t zone_id`	Zone ID - may use `ZbZclIasAceServerGetFreeZoneId`

5.16 IAS WD cluster

```
#include "zcl/security/zcl.ias_wd.h"
```

5.16.1 IAS WD cluster description

Table 308. IAS WD cluster PICS code description

PICS Code	Name	Supported	Notes
IASWD.S	Server	True	-
IASWD.C	Client	True	-
IASWD.S.A0000	`MaxDuration` server attribute	True	-
IASWD.S.Afffd	`ClusterRevision` server attribute	True	-
IASWD.S.Afffe	`AttributeReportingStatus` server attribute	False	-
IASWD.S.C00.Rsp	Start warning server command	True	-
IASWD.S.C01.Rsp	Squawk server command	True	-
IASWD.C.Afffd	`ClusterRevision` client attribute	True	-
IASWD.C.Afffe	`AttributeReportingStatus` client attribute	False	-
IASWD.C.C00.Tx	Start warning client command	True	
IASWD.C.C01.Tx	Squawk client command	True	

5.16.2 IAS WD cluster functions

5.16.2.1 *ZbZclIasWdClientAlloc*

```
struct ZbZclClusterT * ZbZclIasWdClientAlloc(struct ZigBeeT *zb, uint8_t endpoint, void
*arg);
```

Create a new instance of the IAS Warning Device Client cluster

Table 309. ZbZclIasWdClientAlloc parameters

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: Cluster pointer, or NULL if there is an error.

5.16.2.2 *ZbZclIasWdServerAlloc*

```
struct ZbZclClusterT * ZbZclIasWdServerAlloc(struct ZigBeeT *zb, uint8_t endpoint, struct
ZbZclIasWdServerCallbacksT *callbacks, void *arg);
```

Create a new instance of the IAS Warning Device Server cluster

Table 310. **ZbZclIasWdServerAlloc parameters**

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster
`callbacks`	Structure containing any callback function pointers for this cluster
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: Cluster pointer, or NULL if there is an error.

5.16.3 IAS WD cluster enumerations

5.16.3.1 *ZbZclIasWdLevelT*

IAS Warning Device Siren Level Field Values

Table 311. **ZbZclIasWdLevelT parameters**

Parameter	Description
ZCL_IAS_WD_LEVEL_LOW	Low level sound
ZCL_IAS_WD_LEVEL_MEDIUM	Medium level sound
ZCL_IAS_WD_LEVEL_HIGH	High level sound
ZCL_IAS_WD_LEVEL_VERY_HIGH	Very high level sound

5.16.3.2 *ZbZclIasWdSquawkModeT*

IAS Warning Device Squawk Mode Field

Table 312. **ZbZclIasWdSquawkModeT parameters**

Parameter	Description
ZCL_IAS_WD_SQUAWK_MODE_ARMED	Notification sound for “System is armed”
ZCL_IAS_WD_SQUAWK_MODE_DISARMED	Notification sound for "System is disarmed"

5.16.3.3 *ZbZclIasWdStrobeT*

IAS Warning Device Strobe Field

Table 313. **ZbZclIasWdStrobeT parameters**

Parameter	Desription
ZCL_IAS_WD_STROBE_OFF	No strobe
ZCL_IAS_WD_STROBE_ON	Use strobe in parallel to warning

5.16.3.4 *ZbZclIasWdSvrAttrT*

IAS Warning Device server attribute IDs

Table 314. ZbZclIasWdSvrAttrT parameters

Parameter	Description
ZCL_IAS_WD_SVR_ATTR_MAX_DURATION	`MaxDuration`

5.16.3.5 *ZbZclIasWdWarningModeT*

IAS Warning Device Warning Modes

Table 315. ZbZclIasWdWarningModeT parameter

Parameter	Description
ZCL_IAS_WD_WARNING_MODE_STOP	Stop (no warning)
ZCL_IAS_WD_WARNING_MODE_BURGLAR	Burglar
ZCL_IAS_WD_WARNING_MODE_FIRE	Fire
ZCL_IAS_WD_WARNING_MODE_EMERGENCY	Emergency
ZCL_IAS_WD_WARNING_MODE_POLICE_PANIC	Police panic
ZCL_IAS_WD_WARNING_MODE_FIRE_PANIC	Fire panic
ZCL_IAS_WD_WARNING_MODE_EMERGENCY_PANI C	Emergency Panic (i.e., medical issue)

5.16.4 IAS WD cluster structures

5.16.4.1 *ZbZclIasWdClientSquawkReqT*

IAS Warning Device Client Squawk command structure

Table 316. ZbZclIasWdClientSquawkReqT parameters

Parameter	Description
`enum ZbZclIasWdSquawkModeTsquawk_mode`	Squawk mode
`enum ZbZclIasWdStrobeT strobe`	Strobe
`enum ZbZclIasWdLevelTsquawk_level`	Squawk level

5.16.4.2 *ZbZclIasWdClientStartWarningReqT*

IAS Warning Device Client Start Warning command structure

Table 317. ZbZclIasWdClientStartWarningReqT parameters

Parameter	Desription
`enum ZbZclIasWdWarningModeT warning_mode`	Warning Mode
`enum ZbZclIasWdStrobeT strobe`	Strobe
`enum ZbZclIasWdLevelT siren_level`	Siren Level
`uint16_t warning_duration`	Warning Duration
`uint8_t strobe_dutycycle`	Strobe Duty Cycle
`enum ZbZclIasWdLevelTstrobe_level`	Strobe Level

5.16.4.3 *ZbZclIasWdServerCallbacksT*

IAS Warning Server callbacks configuration

Table 318. ZbZclIasWdServerCallbacksT parameters

Parameter	Description
start_warning (callback function pointer)	`uint8_t (*start_warning)(struct ZbZclClusterT *cluster, void *arg,struct ZbZclIasWdClientStartWarningReqT*warn_req)` Callback to handle Start Warning command. ZCL_STATUS_SUCCESS if successful, or other ZclStatusCodeT value on error.
squawk (callback function pointer)	`uint8_t (*squawk)(struct ZbZclClusterT *cluster, void*arg,struct ZbZclIasWdClientSquawkReqT *squawk_req)` Callback to handle Squawk command. ZCL_STATUS_SUCCESS if successful, or other ZclStatusCodeT value on error.

5.17 IAS Zone cluster

```
#include "zcl/security/zcl.ias_zone.h"
```

5.17.1 IAS Zone cluster description

Table 319. IAS zonIAS Zone clustere PICS code description

PICS Code	Name	Supported	Notes
IASZ.S	Server	True	-
IASZ.C	Client	True	-
IASZ.TTP	Trip-to-Pair Enrollment	True	-
IASZ.ARSP	Auto-Enroll-Response Enrollment	True	-
IASZ.AREQ	Auto-Enroll-Request Enrollment	True	-
IASZ.PIXIT01	See Test Spec	False	-
IASZ.PIXIT02	See Text Spec	False	-
IASZ.S.A0000	`ZoneState` server attribute	True	-
IASZ.S.A0001	`ZoneType` server attribute	True	-
IASZ.S.A0002	`ZoneStatus` server attribute	True	-
IASZ.S.A0010	`IAS_CIE_Address` server attribute	True	-
IASZ.S.A0011	`ZoneID` server attribute	True	-
IASZ.S.Afffd	`ClusterRevision` server attribute	True	-
IASZ.S.Afffe	`AttributeReportingStatus` server attribute	False	-
IASZ.S.C00.Rsp	Zone Enroll Response server command	True	-
IASZ.S.C01.Rsp	Initiate Normal Operation Mode server command	True	-
IASZ.S.C02.Rsp	Initiate Test Mode server command	True	-
IASZ.S.C00.Tx	Zone Status Change Notification server command	True	-
IASZ.S.C01.Tx	Zone Enroll Request server command	True	-
IASZ.C.Afffd	`ClusterRevision` client attribute	True	-
IASZ.C.Affe	`AttributeReportingStatus` client attribute	False	-
IASZ.C.C00.Rsp	Zone Status Change Notification client command	True	-

PICS Code	Name	Supported	Notes
IASZ.C.C01.Rsp	Zone Enroll Request client command	True	-
IASZ.C.C00.Tx	Zone Enroll Response client command	True	-
IASZ.C.C01.Tx	Initiate Normal Operation Mode client command	True	-
IASZ.C.C02.Tx	Initiate Test Mode client command	True	-

5.17.2 IAS Zone cluster functions

5.17.2.1 *ZbZclIasZoneClientAlloc*

```
struct ZbZclClusterT * ZbZclIasZoneClientAlloc(struct ZigBeeT *zb, uint8_t endpoint, struct
ZbZclIasZoneClientCallbacksT *callbacks, void *arg);
```

Create a new instance of the IAS Zone Client cluster

Table 320. **ZbZclIasZoneClientAlloc parameters**

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster
`callbacks`	Structure containing any callback function pointers for this cluster
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: Cluster pointer, or NULL if there is an error.

5.17.2.2 *ZbZclIasZoneClientInitiateAutoEnroll*

```
uint8_t ZbZclIasZoneClientInitiateAutoEnroll(struct ZbZclClusterT *cluster, const struct
ZbApsAddrT *dst, void (*callback)(const struct ZbZclWriteRspT *, void *), void *arg);
```

Send a Zone Auto-Enroll request command

Table 321. **ZbZclIasZoneClientInitiateAutoEnroll parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.17.2.3 *ZbZclIasZoneClientInitiateNormalMode*

```
uint8_t ZbZclIasZoneClientInitiateNormalMode(struct ZbZclClusterT *cluster, const struct
ZbApsAddrT *dst, void (*callback)(struct ZbZclCommandRspT *zcl_rsp, void*arg), void *arg);
```

Send a Initiate Normal Operation Mode request command

Table 322. **ZbZclIasZoneClientInitiateNormalMode parameters**

Parameter	Desciption
`cluster`	Cluster instance from which to send this command

Parameter	Desciption
`dst`	Requested destination address
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.17.2.4 *ZbZclIasZoneClientInitiateTestMode*

```
uint8_t ZbZclIasZoneClientInitiateTestMode(struct ZbZclClusterT *cluster, const struct
ZbApsAddrT *dst, struct ZbZclIasZoneClientTestModeReqT *req, void (*callback)(struct
ZbZclCommandRspT *zcl_rsp, void *arg), void *arg);
```

Send a Initiate Test Mode request command

Table 323. ZbZclIasZoneClientInitiateTestMode parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`req`	Initiate Test Mode request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.17.2.5 *ZbZclIasZoneClientSendAutoEnrollResponse*

```
uint8_t ZbZclIasZoneClientSendAutoEnrollResponse(struct ZbZclClusterT *cluster, const struct
ZbApsAddrT *dst, uint8_t zone_id, void (*callback)(struct ZbZclCommandRspT *zcl_rsp, void
*arg), void *arg);
```

Send a Zone Auto-Enroll response command

Table 324. ZbZclIasZoneClientSendAutoEnrollResponse parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`zone_id`	Zone ID of zone in question
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.17.2.6 *ZbZclIasZoneServerAlloc*

```
struct ZbZclClusterT * ZbZclIasZoneServerAlloc(struct ZigBeeT *zb, uint8_t
endpoint, uint16_t zone_type, uint16_t manuf_code, bool use_trip_pair, struct
ZbZclIasZoneServerCallbacksT *callbacks, void *arg);
```

Create a new instance of the IAS Zone Server cluster

Table 325. ZbZclIasZoneServerAlloc parameters

Parameter	Description
zb	Zigbee® stack instance
endpoint	Endpoint on which to create cluster
zone_type	Zone Type
manuf_code	Manufacturer Code
use_trip_pair	If true, use 'trip-to-pair' application must call ZbZclIasZoneServerEnrollRequest to perform the 'trip-to-pair' process, unless the IAS CIE has sent us an unsolicited Auto- Enroll-Response
callbacks	Structure containing any callback function pointers for this cluster
arg	Pointer to application data that is provided back to the callback function when invoked

Return: Cluster pointer, or NULL if there is an error.

5.17.2.7 *ZbZclIasZoneServerEnrollRequest*

```
enum ZclStatusCodeT ZbZclIasZoneServerEnrollRequest(struct ZbZclClusterT *cluster, void
(*callback)(struct ZbZclIasZoneClientEnrollResponseT *enrl_rsp, void *arg), void *arg);
```

Send a Zone Enroll request command Used with 'trip-to-pair'. Before sending a Zone Enroll Request, the IAS CIE must write to the IAS_CIE_Address attribute with its IEEE address.

Table 326. ZbZclIasZoneServerEnrollRequest parameters

Parameter	Description
cluster	Cluster instance from which to send this command
callback	Callback function that is invoked when the response is received
arg	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other ZclStatusCodeT value on error.

5.17.3 IAS Zone cluster enumerations

5.17.3.1 *ZbZclIasZoneClientResponseCodeT*

IAS Zone Enroll Response Code

Table 327. ZbZclIasZoneClientResponseCodeT parameters

Parameter	Description
ZCL_IAS_ZONE_CLI_RESP_SUCCESS	Success
ZCL_IAS_ZONE_CLI_RESP_NOT_SUPPORTED	Not supported
ZCL_IAS_ZONE_CLI_RESP_NO_ENROLL_PERMIT	No enroll permit
ZCL_IAS_ZONE_CLI_RESP_TOO_MANY_ZONES	Too many zones

5.17.3.2 *ZbZclIasZoneServerAttrT*

IAS Zone server attribute IDs

Table 328. ZbZclIasZoneServerAttrT parameters

Parameter	Description
ZCL_IAS_ZONE_SVR_ATTR_ZONE_STATE	Zone State

Parameter	Description
ZCL_IAS_ZONE_SVR_ATTR_ZONE_TYPE	Zone Type
ZCL_IAS_ZONE_SVR_ATTR_ZONE_STATUS	Zone Status
ZCL_IAS_ZONE_SVR_ATTR_CIE_ADDR	IAS_CIE_Address
ZCL_IAS_ZONE_SVR_ATTR_ZONE_ID	Zone ID
ZCL_IAS_ZONE_SVR_ATTR_NUM_ZONE_SENSITIVITY_SUPPORTED	Number Of Zone Sensitivity Levels Supported (optional)
ZCL_IAS_ZONE_SVR_ATTR_CURRENT_ZONE_SENSITIVITY_LEVEL	Current Zone Sensitivity Level (optional)
ZCL_IAS_ZONE_SVR_ATTR_CIE_ENDPOINT	Exegin internal (optional)

5.17.3.3 *ZbZclIasZoneServerModeT*

IAS Zone ZoneStatus Attribute Bit Test Value

Table 329. **ZbZclIasZoneServerModeT parameters**

Parameter	Description
ZCL_IAS_ZONE_SVR_MODE_NORMAL	Normal
ZCL_IAS_ZONE_SVR_MODE_TEST	Set

5.17.3.4 *ZbZclIasZoneServerZoneStateT*

IAS Zone ZoneState Attribute

Table 330. **ZbZclIasZoneServerZoneStateT parameters**

Parameter	Description
ZCL_IAS_ZONE_SVR_STATE_NOT_ENROLLED	Not enrolled
ZCL_IAS_ZONE_SVR_STATE_ENROLLED	Enrolled (the client reacts to Zone State Change Notification commands from the server)

5.17.3.5 *ZbZclIasZoneServerZoneStatusT*

Table 331. **IAS Zone ZoneStatus Attribute**

Parameter	Description
ZCL_IAS_ZONE_SVR_ZONE_STATUS_ALARM1	Alarm 1
ZCL_IAS_ZONE_SVR_ZONE_STATUS_ALARM2	Alarm 2
ZCL_IAS_ZONE_SVR_ZONE_STATUS_TAMPER	Tamper
ZCL_IAS_ZONE_SVR_ZONE_STATUS_BATTERY	Battery
ZCL_IAS_ZONE_SVR_ZONE_STATUS_SUPERVISION _REPORTS	Supervision Notify
ZCL_IAS_ZONE_SVR_ZONE_STATUS_RESTORE_RE PORTS	Restore Notify
ZCL_IAS_ZONE_SVR_ZONE_STATUS_TROUBLE	Trouble
ZCL_IAS_ZONE_SVR_ZONE_STATUS_AC_MAINS	AC (mains)
ZCL_IAS_ZONE_SVR_ZONE_STATUS_TEST	Test
ZCL_IAS_ZONE_SVR_ZONE_STATUS_BATTERY_DE FECT	Battery Defect

5.17.3.6 *ZbZclIasZoneServerZoneTypeT*

Table 332. **IAS Zone ZoneType Attribute**

Attributes	Description
ZCL_IAS_ZONE_SVR_ZONE_TYPE_STANDARD_CIE	Standard CIE
ZCL_IAS_ZONE_SVR_ZONE_TYPE_MOTION_SENSO R	Motion sensor
ZCL_IAS_ZONE_SVR_ZONE_TYPE_CONTACT_SWIT CH	Contact switch
ZCL_IAS_ZONE_SVR_ZONE_TYPE_DOOR_WINDOW	Door/Window handle
ZCL_IAS_ZONE_SVR_ZONE_TYPE_FIRE_SENSOR	Fire sensor
ZCL_IAS_ZONE_SVR_ZONE_TYPE_WATER_SENSOR	Water sensor
ZCL_IAS_ZONE_SVR_ZONE_TYPE_CO_SENSOR	Carbon Monoxide (CO) sensor
ZCL_IAS_ZONE_SVR_ZONE_TYPE_PERSONAL_EME RG_DEVICE	Personal emergency device
ZCL_IAS_ZONE_SVR_ZONE_TYPE_MOVEMENT_SE NSOR	Vibration/Movement sensor
ZCL_IAS_ZONE_SVR_ZONE_TYPE_REMOTE_CONT ROL	Remote Control
ZCL_IAS_ZONE_SVR_ZONE_TYPE_KEY_FOB	Key fob
ZCL_IAS_ZONE_SVR_ZONE_TYPE_KEYPAD	Keypad
ZCL_IAS_ZONE_SVR_ZONE_TYPE_STANDARD_WA RNING_DEVICE	Standard Warning Device (see [N1] part 4)
ZCL_IAS_ZONE_SVR_ZONE_TYPE_GLASS_SENSOR	Glass break sensor
ZCL_IAS_ZONE_SVR_ZONE_TYPE_SECURITY_REPE ATER	Security repeater
ZCL_IAS_ZONE_SVR_ZONE_TYPE_INVALID	Invalid Zone Type

5.17.4 IAS Zone cluster structures

5.17.4.1 *ZbZclIasZoneClientCallbacksT*

IAS Zone Client callbacks configuration

Table 333. **ZbZclIasZoneClientCallbacksT parameters**

Parameter	Description
zone_status_change (callback function pointer)	`Void (zone_status_change)(struct ZbZclClusterT*cluster,void *arg, / ZbZclClusterSetCallbackArg*/struct ZbZclIasZoneServerStatusChangeNotifyT*notify,const struct ZbApsAddrT *src)` Callback to application, invoked on receipt of Zone Status Change Notification command
zone_enroll_req (callback function pointer)	`uint8_t (zone_enroll_req)(struct ZbZclClusterT*cluster,void *arg, / ZbZclClusterSetCallbackArg*/struct ZbZclIasZoneServerEnrollRequestT *req,uint64_t ext_src_addr,enum ZbZclIasZoneClientResponseCodeT*rsp_code,uint8_t *zone_id)` Callback to application, invoked on receipt of Zone Enroll command

5.17.4.2 *ZbZclIasZoneClientEnrollResponseT*

Zone Enroll response structure

Table 334. **ZbZclIasZoneClientEnrollResponseT parameters**

Parameter	Description
`uint8_t zcl_status`	Response status

Parameter	Description
`enum ZbZclIasZoneClientResponseCodeT enroll_status`	Enroll response code
`uint8_t zone_id`	Zone ID

5.17.4.3 *ZbZclIasZoneClientTestModeReqT*

Initiate Test Mode request structure

Table 335. ZbZclIasZoneClientTestModeReqT parameters

Parameter	Description
`uint8_t test_duration`	Test Mode Duration
`uint8_t current_zone_sensitivity`	Current Zone Sensitivity Level

5.17.4.4 *ZbZclIasZoneServerCallbacksT*

IAS Zone Server callbacks configuration

Table 336. ZbZclIasZoneServerCallbacksT parameters

Parameter	Description
mode_change (callback function pointer)	`uint8_t (mode_change)(struct ZbZclClusterT *cluster,void*arg, / ZbZclClusterSetCallbackArg */enum ZbZclIasZoneServerModeTmode,struct ZbZclIasZoneClientTestModeReqT *req)` Callback to application, invoked on receipt of Initiate Normal Operation Mode or Initiate Test Mode command

5.17.4.5 *ZbZclIasZoneServerEnrollRequestT*

Zone Enroll request structure

Table 337. ZbZclIasZoneServerEnrollRequestT parameters

Parameter	Description
`uint16_t zone_type`	Zone Type
`uint16_t manuf_code`	Manufacturer Code

5.17.4.6 *ZbZclIasZoneServerStatusChangeNotifyT*

Zone State Change Notification request structure

Table 338. ZbZclIasZoneServerStatusChangeNotifyT parameters

Parameter	Description
`enum ZbZclIasZoneServerZoneStatusT zone_status`	Zone Status
`uint8_t ext_status`	Extended Status
`uint8_t zone_id`	Zone ID
`uint16_t delay`	Delay

5.18 Identify cluster

```
#include "zcl/general/zcl.identify.h"
```

5.18.1 Identify cluster description

Table 339. **Identify cluster PISC code description**

PICS Code	Name	Supported	Notes
I.S	Server	True	-
I.C	Client	True	-
I.S.A0000	`IdentifyTime` server attribute	True	-
I.S.Afffd	`ClusterRevision` server attribute	True	-
I.S.C00.Rsp	Identify server command	True	-
I.S.C01.Rsp	Identify Query server command	True	-
I.S.C40.Rsp	Trigger effect server command	False	-
I.S.C00.Tx	Identify Query Response server command	True	-
I.C.Afffd	`ClusterRevision` client attribute	True	-
I.C.C00.Rsp	Identify Query Response client command	True	-
I.C.C00.Tx	Identify client command	True	-
I.C.C01.Tx	Identify Query client command	True	-
I.C.C40.Tx	Trigger effect client command	False	-

5.18.2 Identify cluster functions

5.18.2.1 ZbZclIdentifyClientAlloc

```
struct ZbZclClusterT * ZbZclIdentifyClientAlloc(struct ZigBeeT *zb, uint8_t endpoint);
```

Create a new instance of the Identify Client cluster

Table 340. **ZbZclIdentifyClientAlloc parameters**

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster

Return: Cluster pointer, or NULL if there is an error.

5.18.2.2 ZbZclIdentifyServerAlloc

```
struct ZbZclClusterT * ZbZclIdentifyServerAlloc(struct ZigBeeT *zb, uint8_t endpoint, void
*arg);
```

Create a new instance of the Identify Server cluster

Table 341. **ZbZclIdentifyServerAlloc parameters**

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster

Parameter	Description
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: Cluster pointer, or NULL if there is an error.

5.18.2.3 *ZbZclIdentifyServerGetTime*

```
uint16_t ZbZclIdentifyServerGetTime(struct ZbZclClusterT *cluster);
```

Get the local Identify Server time

Table 342. **ZbZclIdentifyServerGetTime parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command

Return: uint16_t Time remaining in zigbee timer

5.18.2.4 *ZbZclIdentifyServerSetCallback*

```
void ZbZclIdentifyServerSetCallback(struct ZbZclClusterT *cluster, ZbZclIdentifyCallbackT
callback);
```

Set the callback in the cluster private structure.

Table 343. **ZbZclIdentifyServerSetCallback parameters**

Parameter	Descption
`cluster`	Cluster instance from which to send this command
`callback`	Callback function that is invoked when the response is received

Return: Void

5.18.2.5 *ZbZclIdentifyServerSetTime*

```
void ZbZclIdentifyServerSetTime(struct ZbZclClusterT *cluster, uint16_t seconds);
```

Set the local Identify Server time

If BDB_COMMISSION_MODE_FIND_BIND is enabled and seconds > 0, seconds is adjusted to be >= ZB_BDBC_MinCommissioningTime

Table 344. **ZbZclIdentifyServerSetTime parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`seconds`	Seconds for updating the identify time counter

Return: Void

5.18.2.6 *zcl_identify_identify_request*

```
enum ZclStatusCodeT zcl_identify_identify_request(struct ZbZclClusterT *cluster, const
struct ZbApsAddrT *dst, uint16_t identify_time, void (*callback)(struct ZbZclCommandRspT
*rsp, void *arg), void *arg);
```

Send an Identify command

Table 345. zcl_identify_identify_request parameters

Parameter	Description
cluster	Cluster instance from which to send this command
dst	Requested destination address
identify_time	Time which is used to set the IdentifyTime attribute
callback	Callback function that is invoked when the response is received
arg	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other ZclStatusCodeT value on error.

5.18.2.7 *zcl_identify_query_request*

```
enum ZclStatusCodeT zcl_identify_query_request(struct ZbZclClusterT *cluster, const struct
ZbApsAddrT *dst, void (*callback)(struct ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send an Identify Query command

Table 346. zcl_identify_query_request parameters

Parameter	Description
cluster	Cluster instance from which to send this command
dst	Requested destination address
callback	Callback function that is invoked when the response is received
arg	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other ZclStatusCodeT value on error.

5.18.3 Identify cluster numerations

5.18.3.1 *ZbZclIdentifySvrAttrT*

Identify server attribute IDs

Table 347. ZbZclIdentifySvrAttrT paramters

Parameter	Description
ZCL_IDENTIFY_ATTR_TIME	Identify Time

5.19 Illuminance level sensing cluster

```
#include "zcl/general/zcl.illum.level.h"
```

5.19.1 Illuminance level sensing cluster description

Table 348. Illuminance level sensing cluster PICS code description

PICS Code	Name	Supported	Notes
ILLVL.S	Server	True	-
ILLVL.C	Client	True	-
ILLVL.S.A0000	LevelStatus server attribute	True	-

PICS Code	Name	Supported	Notes
ILLVL.S.A0010	`IlluminanceTargetLevel` server attribute	True	-
ILLVL.S.Afffd	`ClusterRevision` server attribute	True	-
ILLVL.S.Afffe	`AttributeReportingStatus` server attribute	False	-
ILLVL.C.Afffd	`ClusterRevision` client attribute	True	-
ILLVL.C.Afffe	`AttributeReportingStatus` client attribute	False	-

5.19.2 Illuminance level sensing cluster functions

5.19.2.1 ZbZclIllumLevelClientAlloc

```
struct ZbZclClusterT * ZbZclIllumLevelClientAlloc(struct ZigBeeT *zb, uint8_t endpoint);
```

Create a new instance of the Illuminance Level Sensing Client cluster

Table 349. ZbZclIllumLevelClientAlloc parameters

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster

Return: Cluster pointer, or NULL if there is an error.

5.19.2.2 ZbZclIllumLevelServerAlloc

```
struct ZbZclClusterT * ZbZclIllumLevelServerAlloc(struct ZigBeeT *zb, uint8_t endpoint);
```

Create a new instance of the Illuminance Level Sensing Server cluster

Table 350. ZbZclIllumLevelServerAlloc parameters

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster

Return: Cluster pointer, or NULL if there is an error.

5.19.3 Illuminance level sensing cluster enumerations

5.19.3.1 ZbZclIllumLevelSvrAttrT

Table 351. ZbZclIllumLevelSvrAttrT attriute identifiers

Parameter	Description
ZCL_ILLUM_LEVEL_ATTR_LEVEL_STATUS	Level Status
ZCL_ILLUM_LEVEL_ATTR_LIGHT_SENSOR_TYPE	Light Sensor Type
ZCL_ILLUM_LEVEL_ATTR_ILLUM_TARGET_LEVEL	Illuminance Target Level

5.20 Illuminance measurement cluster

```
#include "zcl/general/zcl.illum.meas.h"
```

5.20.1 Illuminance measurement cluster description

Table 352. Illuminance measurement cluster PICS code description

PICS Code	Name	Supported	Notes
IM.S	Server	True	-
IM.C	Client	True	-
IM.S.A0000	`MeasuredValue` server attribute	True	-
IM.S.A0000.Report.Tx	`MeasuredValue` server attribute reports	True	-
IM.S.A0001	`MinMeasuredValue` server attribute	True	-
IM.S.A0002	`MaxMeasuredValue` server attribute	True	-
IM.S.Afffd	`ClusterRevision` server attribute	True	-
IM.S.Afffe	`AttributeReportingStatus` server attribute	False	-
IM.C.A0000.Report.Rsp	MeasuredValue client attribute reports	False	-
IM.C.Afffd	`ClusterRevision` client attribute	True	-
IM.C.Afffe	`AttributeReportingStatus` client attribute	False	-

5.20.2 Illuminance measurement cluster functions

5.20.2.1 *ZbZclIllumMeasClientAlloc*

```
struct ZbZclClusterT * ZbZclIllumMeasClientAlloc(struct ZigBeeT *zb, uint8_t endpoint);
```

Create a new instance of the Illuminance Measurement Client cluster

Table 353. ZbZclIllumMeasClientAlloc parameters

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster

Return: Cluster pointer, or NULL if there is an error.

5.20.2.2 *ZbZclIllumMeasServerAlloc*

```
struct ZbZclClusterT * ZbZclIllumMeasServerAlloc(struct ZigBeeT *zb, uint8_t endpoint,
uint16_t min, uint16_t max);
```

Create a new instance of the Illuminance Measurement Server cluster

Table 354. ZbZclIllumMeasServerAlloc parameters

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster
`min`	Minimum value capable of being measured (MinMeasuredValue)
`max`	Maximum value capable of being measured (MaxMeasuredValue)

Return: Cluster pointer, or NULL if there is an error.

5.20.3 Illuminance measurement cluster enumerations

5.20.3.1 ZbZclIllumMeasSvrAttrT

Illuminance Measurement server attribute IDs

Table 355. ZZclIllumMeasSvrAttrT attributes

Attribute	Description
ZCL_ILLUM_MEAS_ATTR_MEAS_VAL	Measured Value
ZCL_ILLUM_MEAS_ATTR_MIN_MEAS_VAL	Min Measured Value
ZCL_ILLUM_MEAS_ATTR_MAX_MEAS_VAL	Max Measured Value
ZCL_ILLUM_MEAS_ATTR_TOLERANCE	Tolerance (optional)
ZCL_ILLUM_MEAS_ATTR_LIGHT_SENSOR_TYPE	Light Sensor Type (optional)

5.21 Level cluster

```
#include "zcl/general/zcl.level.h"
```

5.21.1 Level cluster description

Table 356. Level cluster PICS code descript

PICS Code	Name	Supported	Notes
LC.S	Server	True	-
LC.C	Client	True	-
LC.S.A0000	`CurrentLevel` server attribute	True	-
LC.S.A0000.Scene	`CurrentLevel` server attribute scenes	True	-
LC.S.A0000.Report.Tx	`CurrentLevel` server attribute reports	True	-
LC.S.A000f	`OnOffTransitionTime` server attribute	True	Optional
LC.S.Afffd	`ClusterRevision` server attribute	True	-
LC.S.C00.Rsp	Move to Level server command	True	-
LC.S.C01.Rsp	Move server command	True	-
LC.S.C02.Rsp	Step server command	True	-
LC.S.C03.Rsp	Stop server command	True	-
LC.S.C04.Rsp	Move to Level (with On/Off) server command	True	-
LC.S.C05.Rsp	Move (with On/Off) server command	True	-
LC.S.C06.Rsp	Step (with On/Off) server command	True	-
LC.S.C07.Rsp	Stop server command	True	-
LC.C.A0000.Report.Rsp	`CurrentLevel` client attribute reports	False	-
LC.C.Afffd	`ClusterRevision` client attribute	True	-
LC.S.C00.Tx	Move to Level server command	True	-
LC.S.C01.Tx	Move server command	True	-
LC.S.C02.Tx	Step server command	True	-
LC.S.C03.Tx	Stop server command	True	-
LC.S.C04.Tx	Move to Level (with On/Off) server command	True	-
LC.S.C05.Tx	Move (with On/Off) server command	True	-

PICS Code	Name	Supported	Notes
LC.S.C06.Tx	Step (with On/Off) server command	True	-
LC.S.C07.Tx	Stop server command	True	-

5.21.2 Level cluster functions

5.21.2.1 *ZbZclLevelClientAlloc*

```
struct ZbZclClusterT * ZbZclLevelClientAlloc(struct ZigBeeT *zb, uint8_t endpoint);
```

Create a new instance of the Level Client cluster

Table 357. ZbZclLevelClientAlloc parameters

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster

Return: Cluster pointer, or NULL if there is an error.

5.21.2.2 *ZbZclLevelClientMoveReq*

```
enum ZaclStatusCodeT ZbZclLevelClientMoveReq(struct ZbZclClusterT *cluster, const
struct ZbApsAddrT *dst, struct ZbZclLevelClientMoveReqT *req, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Move command

Table 358. ZbZclLevelClientMoveReq parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`req`	Move command request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.21.2.3 *ZbZclLevelClientMoveToLevelReq*

```
enum ZclStatusCodeT ZbZclLevelClientMoveToLevelReq(struct ZbZclClusterT *cluster, const
struct ZbApsAddrT *dst, struct ZbZclLevelClientMoveToLevelReqT *req, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Move to Level command

Table 359. ZbZclLevelClientMoveToLevelReq parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`req`	Move To Level command request structure

Parameter	Description
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.21.2.4 *ZbZclLevelClientStepReq*

```
enum ZclStatusCodeT ZbZclLevelClientStepReq(struct ZbZclClusterT *cluster, const struct
ZbApsAddrT *dst, struct ZbZclLevelClientStepReqT *req, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Step command

Table 360. **ZbZclLevelClientStepReq parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`req`	Step command request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.21.2.5 *ZbZclLevelClientStopReq*

```
enum ZclStatusCodeT ZbZclLevelClientStopReq(struct ZbZclClusterT *cluster, const struct
ZbApsAddrT *dst, struct ZbZclLevelClientStopReqT *req, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Stop command

Table 361. **ZbZclLevelClientStopReq parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`req`	Stop command request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.21.2.6 *ZbZclLevelServerAlloc*

```
struct ZbZclClusterT * ZbZclLevelServerAlloc(struct ZigBeeT *zb, uint8_t endpoint, struct
ZbZclClusterT *onoff_server, struct ZbZclLevelServerCallbacksT *callbacks, void *arg);
```

Create a new instance of the Level Server cluster

Table 362. ZbZclLevelServerAlloc parameters

Parameter	Description
zb	Zigbee® stack instance
endpoint	Endpoint on which to create cluster
onoff_server	OnOff server cluster pointer for processing commands with the Options fields. May be NULL
callbacks	Structure containing any callback function pointers for this cluster
arg	Pointer to application data that is provided back to the callback function when invoked

Return: Cluster pointer, or NULL if there is an error.

5.21.3 Level cluster enumerations

5.21.3.1 *ZbZclLevelSvrAttrT*

Level server attribute IDs

Table 363. ZbZlLevelSvrAttrT attribute IDs

Attributes	Description
ZCL_LEVEL_ATTR_CURRLEVEL	Current Level
ZCL_LEVEL_ATTR_REMAINTIME	Remaining Time (optional)
ZCL_LEVEL_ATTR_MINLEVEL	Min Level (optional)
ZCL_LEVEL_ATTR_MAXLEVEL	Max Level (optional)
ZCL_LEVEL_ATTR_CURRFREQ	Current Frequency (optional)
ZCL_LEVEL_ATTR_MINFREQ	Min Frequency (optional)
ZCL_LEVEL_ATTR_MAXFREQ	Max Frequency (optional)
ZCL_LEVEL_ATTR_OPTIONS	On Off Transition Time (optional)
ZCL_LEVEL_ATTR_ONOFF_TRANS_TIME	On Level (optional)
ZCL_LEVEL_ATTR_ONLEVEL	On Transition Time (optional)
ZCL_LEVEL_ATTR_ON_TRANS_TIME	Off Transition Time (optional)
ZCL_LEVEL_ATTR_OFF_TRANS_TIME	Default Move Rate (optional)
ZCL_LEVEL_ATTR_DEFAULT_MOVE_RATE	Options (optional)
ZCL_LEVEL_ATTR_STARTUP_CURRLEVEL	Start Up Current Level (optional)

5.21.4 Level cluster structures

5.21.4.1 *ZbZclLevelClientMoveReqT*

Move command structure `ZbZclLevelClientMoveReqT`

Table 364. ZbZclLevelClientMoveReqT parameters

Parameter	Description
bool with_onoff	With Onoff - If true then cmd is ZCL_LEVEL_COMMAND_STOP_ONOFF, else ZCL_LEVEL_COMMAND_STOP
uint8_t mode	Mode
uint8_t rate	Rate
uint8_t mask	Options mask - Not included if with_onoff is true

Parameter	Description
`uint8_t override`	Options override - Not included if with_onoff is true

5.21.4.2 *ZbZclLevelClientMoveToLevelReqT*

Move To Level command structure

Table 365. ZbZclLevelClientMoveToLevelReqT parameters

Parameter	Description
`bool with_onoff`	With Onoff - If true then cmd is ZCL_LEVEL_COMMAND_STOP_ONOFF, else ZCL_LEVEL_COMMAND_STOP
`uint8_t level`	Level
`uint16_t transition_time`	Transition time
`uint8_t mask`	Options mask - Not included if with_onoff is true
`uint8_t override`	Options override - Not included if with_onoff is true

5.21.4.3 *ZbZclLevelClientStepReqT*

Step command structure

Table 366. ZbZclLevelClientStepReqT parameters

Parameter	Description
`bool with_onoff`	With Onoff - If true then cmd is ZCL_LEVEL_COMMAND_STOP_ONOFF, else ZCL_LEVEL_COMMAND_STOP
`uint8_t mode`	Mode
`uint8_t size`	Size
`uint16_t transition_time`	Transition time
`uint8_t mask`	Options mask - Not included if with_onoff is true
`uint8_t override`	Options override - Not included if with_onoff is true

5.21.4.4 *ZbZclLevelClientStopReqT*

Stop command structure

Table 367. ZbZclLevelClientStopReqT parameters

Parameter	Description
`bool with_onoff`	With Onoff - If true then cmd is ZCL_LEVEL_COMMAND_STOP_ONOFF, else ZCL_LEVEL_COMMAND_STOP
`uint8_t mask`	Options mask - Not included if with_onoff is true
`uint8_t override`	Options override - Not included if with_onoff is true

5.21.4.5 *ZbZclLevelServerCallbacksT*

Level Server callbacks configuration

Table 368. **ZbZclLevelServerCallbacksT parameters**

Parameter	Description
move_to_level (callback function pointer)	`enum ZclStatusCodeT (*move_to_level)(struct ZbZclClusterT *cluster,structZbZclLevelClientMoveToLevelReqT *req, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of Move To Level command. Set with_onoff to true in the req struct when utilizing the On/Off cluster cluster on the same endpoint. The application is expected to update ZCL_LEVEL_ATTR_CURRLEVEL
move (callback function pointer)	`enum ZclStatusCodeT (*move)(struct ZbZclClusterT*cluster,struct ZbZclLevelClientMoveReqT *req, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of Move command. Set with_onoff to true in the req struct when utilizing the On/Off cluster cluster on the same endpoint. The application is expected to update ZCL_LEVEL_ATTR_CURRLEVEL
step (callback function pointer)	`enum ZclStatusCodeT (*step)(struct ZbZclClusterT*cluster,struct ZbZclLevelClientStepReqT *req, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of Step command. Set with_onoff to true in the req struct when utilizing the On/Off cluster cluster on the same endpoint. The application is expected to update ZCL_LEVEL_ATTR_CURRLEVEL
stop (callback function pointer)	`enum ZclStatusCodeT (*stop)(struct ZbZclClusterT*cluster,struct ZbZclLevelClientStopReqT *req, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of Stop command. Set with_onoff to true in the req struct when utilizing the On/Off cluster cluster on the same endpoint. The application is expected to update ZCL_LEVEL_ATTR_CURRLEVEL

5.22 Messaging

```
#include "zcl/se/zcl.message.h"
```

5.22.1 Messaging description

Table 369. **Messaging PISC code description**

PICS Code	Name	Supported	Notes
SEMS.S	Server	True	-
SEMS.C	Client	True	-
SEMS.S.Afffd	`ClusterRevision` server attribute	True	-
SEMS.S.Afffe	`AttributeReportingStatus` server attribute	False	-
SEMS.S.C00.Rsp	Get Last Message server command	False	-
SEMS.S.C01.Rsp	Message Confirmation server command	False	-
SEMS.S.C02.Rsp	`GetMessageCancellation` server command	False	-
SEMS.S.C00.Tx	Display Message server command	True	-
SEMS.S.C01.Tx	Cancel Message server command	True	-
SEMS.S.C02.Tx	Display Protected Message server command	True	-
SEMS.S.C03.Tx	Cancel All Messages server command	True	-
SEMS.C.Afffd	`ClusterRevision` client attribute	True	-
SEMS.C.Afffe	`AttributeReportingStatus` client attribute	False	-
SEMS.C.C00.Rsp	Display Message client command	False	-
SEMS.C.C01.Rsp	Cancel Message client command	False	-

PICS Code	Name	Supported	Notes
SEMS.C.C02.Rsp	Display Protected Message client command	False	-
SEMS.C.C03.Rsp	Cancel All Messages client command	False	-
SEMS.C.C00.Tx	Get Last Message client command	True	-
SEMS.C.C01.Tx	Message Confirmation client command	True	-
SEMS.C.C02.Tx	`GetMessageCancellation` client command	True	-

5.22.2 Messaging functions

5.22.2.1 *ZbZclMsgClientAlloc*

```
struct ZbZclClusterT * ZbZclMsgClientAlloc(struct ZigBeeT *zb, uint8_t endpoint, struct
ZbZclMsgClientCallbacksT *callbacks, void *arg);
```

Create a new instance of the Messaging Client cluster

Table 370. **ZbZclMsgClientAlloc parameters**

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster
`callbacks`	Structure containing any callback function pointers for this cluster
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: Cluster pointer, or NULL if there is an error.

5.22.2.2 *ZbZclMsgClientConfReq*

```
enum ZclStatusCodeT ZbZclMsgClientConfReq(struct ZbZclClusterT *cluster, const struct
ZbApsAddrT *dst, struct ZbZclMsgConfirmT *msg_conf, void (*callback)(struct ZbZclCommandRspT
*rsp, void *arg), void *arg);
```

Send a Message Confirmation command

Table 371. **ZbZclMsgClientConfReq parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`msg_conf`	Message Confirmation command request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.22.2.3 *ZbZclMsgClientGetLastReq*

```
enum ZclStatusCodeT ZbZclMsgClientGetLastReq(struct ZbZclClusterT *cluster, const struct
ZbApsAddrT *dst, void (*callback)(struct ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Get Last Message command

Table 372. **ZbZclMsgClientGetLastReq parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.22.2.4 *ZbZclMsgClientGetMsgCancelReq*

```
enum ZclStatusCodeT ZbZclMsgClientGetMsgCancelReq(struct ZbZclClusterT *cluster, const
struct ZbApsAddrT *dst, uint32_t earliestTime, void (*callback)(struct ZbZclCommandRspT
*rsp, void *arg), void *arg);
```

Send a Get Message Cancellation command

Table 373. **ZbZclMsgClientGetMsgCancelReq parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`earliestTime`	UTC Timestamp indicating the earliest implementation time of a Cancel All Messages command to be returned
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.22.2.5 *ZbZclMsgServerAlloc*

```
struct ZbZclClusterT * ZbZclMsgServerAlloc(struct ZigBeeT *zb, uint8_t endpoint, struct
ZbZclMsgServerCallbacksT *callbacks, void *arg);
```

Create a new instance of the Messaging Server cluster

Table 374. **ZbZclMsgServerAlloc parameters**

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster
`callbacks`	Structure containing any callback function pointers for this cluster
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: Cluster pointer, or NULL if there is an error.

5.22.2.6 *ZbZclMsgServerCancelAllReq*

```
enum ZclStatusCodeT ZbZclMsgServerCancelAllReq(struct ZbZclClusterT *cluster, const struct
ZbApsAddrT *dst, struct ZbZclMsgMessageCancelAllT *cancel_all, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Cancel All Messages command

Table 375. **ZbZclMsgServerCancelAllReq parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`cancel_all`	Cancel All Messages command request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.22.2.7 *ZbZclMsgServerCancelMessageReq*

```
enum ZclStatusCodeT ZbZclMsgServerCancelMessageReq(struct ZbZclClusterT *cluster, const
struct ZbApsAddrT *dst, struct ZbZclMsgMessageCancelT *cancel, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Cancel Message command

Table 376. **ZbZclMsgServerCancelMessageReq parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`cancel`	Cancel Message command request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.22.2.8 *ZbZclMsgServerDisplayMessageReq*

```
enum ZclStatusCodeT ZbZclMsgServerDisplayMessageReq(struct ZbZclClusterT *cluster, const
struct ZbApsAddrT *dst, struct ZbZclMsgMessageT *msg, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Display Message command

Table 377. **ZbZclMsgServerDisplayMessageReq parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`msg`	Display Message command request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.22.2.9 *ZbZclMsgServerDisplayProtectedMsgReq*

```
enum ZclStatusCodeT ZbZclMsgServerDisplayProtectedMsgReq(struct ZbZclClusterT*cluster,
const struct ZbApsAddrT *dst, struct ZbZclMsgMessageT *msg, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Display Protected Message command

Table 378. **ZbZclMsgServerDisplayProtectedMsgReq parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`msg`	Display Protected Message command request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.22.3 Messaging structures

5.22.3.1 *ZbZclMsgClientCallbacksT*

Messaging Client callbacks configuration

Table 379. **ZbZclMsgClientCallbacksT parameters**

Parameter	Description
`display_message` (callback function pointer)	`Enum ZclStatusCodeT (*display_message)(structZbZclClusterT *cluster, void *arg,struct ZbZclMsgMessageT *msg, struct ZbZclAddrInfoT *srcInfo)` Callback to application, invoked on receipt of Display Message command
`cancel_message` (callback function pointer)	`enumZclStatusCodeT (*cancel_message)(structZbZclClusterT *cluster,void*arg,struct ZbZclMsgMessageCancelT *cancel, struct ZbZclAddrInfoT*srcInfo)` Callback to application, invoked on receipt of Cancel Message command
`cancel_all_messages` (callback function pointer)	`enumZclStatusCodeT (*cancel_all_messages)(structZbZclClusterT *cluster, void *arg,structZbZclMsgMessageCancelAllT *cancel_all, struct ZbZclAddrInfoT *srcInfo)` Callback to application, invoked on receipt of Cancel All Messages command
`display_protected_message` (callback function pointer)	`enum ZclStatusCodeT (*display_protected_message)(struct ZbZclClusterT *cluster, void *arg,struct ZbZclMsgMessageT *msg, struct ZbZclAddrInfoT *srcInfo)` Callback to application, invoked on receipt of Display Protected Message command

5.22.3.2 *ZbZclMsgConfirmEnhT*

Enhanced Message Confirmation command structure

Table 380. **ZbZclMsgConfirmEnhT parameters**

Parameter	Description
`uint32_t message_id`	Message ID

Parameter	Description
`uint32_t confirm_time`	Confirmation Time
`uint8_t confirm_control`	Message Confirmation Control
`char confirm_response`	Message confirmation response

5.22.3.3 *ZbZclMsgConfirmT*

Message Confirmation command structure

Table 381. **ZbZclMsgConfirmT parameters**

Parameter	Description
`uint32_t message_id`	Message ID
`uint32_t confirm_time`	Confirmation Time

5.22.3.4 *ZbZclMsgGetMsgCancellationT*

Get Message Cancellation command structure

Table 382. **ZbZclMsgGetMsgCancellationT parameters**

Parameter	Description
`uint32_t earliest_impl_time`	Earliest implementation time

5.22.3.5 *ZbZclMsgMessageCancelAllT*

Cancel All Messages command structure

Table 383. **ZbZclMsgMessageCancelAllT parameters**

Parameter	Description
`uint32_t implementation_time`	Implementation date / time

5.22.3.6 *ZbZclMsgMessageCancelT*

Cancel Message command structure

Table 384. **ZbZclMsgMessageCancelT parameters**

Parameter	Description
`uint32_t message_id`	Message ID
`uint8_t control`	Message Control

5.22.3.7 *ZbZclMsgMessageConfT*

Message Confirmation callback structure

Table 385. **ZbZclMsgMessageConfT parameters**

Parameter	Description
`uint32_t message_id`	Message ID
`uint32_t confirm_time`	Confirmation Time
`bool has_confirm_control`	Has message confirmation control

Parameter	Description
`uint8_t confirm_control`	Message Confirmation Control
`bool has_confirm_response`	Has message confirmation response
`uint8_t confirm_response`	Message confirmation response

5.22.3.8 *ZbZclMsgMessageT*

Display Message/Display Protected Message command structure

Table 386. ZbZclMsgMessageT parameters

Parameter	Description
`uint32_t message_id`	Message ID
`uint32_t start_time`	Start time - UTC time
`uint16_t duration`	Duration in minutes
`uint8_t message_control`	Message control
`char message_str`	Message - UTF-8
`uint8_t extended_control`	Extended message montrol

5.22.3.9 *ZbZclMsgServerCallbacksT*

Messaging Server callbacks configuration

Table 387. ZbZclMsgServerCallbacksT parameters

Parameter	Description
`get_last_message` (callback function pointer)	`Enum clStatusCodeT (*get_last_message)(struct ZbZclClusterT *cluster, void *arg,struct ZbZclAddrInfoT *srcInfo)` Callback to application, invoked on receipt of Get Last Message command
`message_confirmation` (callback function pointer)	`Enum ZclStatusCodeT (*message_confirmation)(struct ZbZclClusterT *cluster, void *arg,struct ZbZclMsgMessageConfT *conf, struct ZbZclAddrInfoT *srcInfo)` Callback to application, invoked on receipt of Message Confirmation command
`get_message_cancellation` (callback function pointer)	`Enum ZclStatusCodeT (*get_message_cancellation)(struct ZbZclClusterT *cluster, void *arg,struct ZbZclMsgGetMsgCancellationT *req, struct ZbZclAddrInfoT *source)` Callback to application, invoked on receipt of Get Message Cancellation command

5.23 Metering cluster

```
#include "zcl/se/zcl.meter.h"
```

5.23.1 Metering cluster description

Table 388. Metering cluster PICS code description

PICS Code	Name	Supported	Notes
SEMT.S	Server	True	-
SEMT.C	Client	True	-

PICS Code	Name	Supported	Notes
SEMT.S.A0000	`CurrentSummationDelivered` server attribute	True	-
SEMT.S.A0200	`Status` server attribute	True	-
SEMT.S.A0300	`UnitofMeasure` server attribute	True	-
SEMT.S.A0303	`SummationFormatting` server attribute	True	Optional
SEMT.S.A0306	`MeteringDeviceType` server attribute	True	-
SEMT.S.Afffd	`ClusterRevision` server attribute	True	-
SEMT.S.Afffe	`AttributeReportingStatus` server attribute	False	-
SEMT.S.C00.Rsp	Get Profile server command	False	-
SEMT.S.C01.Rsp	Request Mirror Response server command	False	-
SEMT.S.C02.Rsp	Mirror Removed server command	False	-
SEMT.S.C03.Rsp	Request Fast Poll Mode server command	False	-
SEMT.S.C04.Rsp	`ScheduleSnapshot` server command	False	-
SEMT.S.C05.Rsp	`TakeSnapshot` server command	False	-
SEMT.S.C06.Rsp	`GetSnapshot` server command	False	-
SEMT.S.C07.Rsp	`StartSampling` server command	False	-
SEMT.S.C08.Rsp	`GetSampledData` server command	False	-
SEMT.S.C09.Rsp	`MirrorReportAttributeResponse` server command	False	-
SEMT.S.C0a.Rsp	ResetLoadLimit Counter server command	False	-
SEMT.S.C0b.Rsp	Change Supply server command	False	-
SEMT.S.C0c.Rsp	Local Change Supply server command	False	-
SEMT.S.C0d.Rsp	`SetSupplyStatus` server command	False	-
SEMT.S.C0e.Rsp	`SetUncontrolledFlowThreshold` server command	False	-
SEMT.S.C00.Tx	Get Profile Response server command	True	-
SEMT.S.C01.Tx	Request Mirror server command	False	-
SEMT.S.C02.Tx	Remove Mirror server command	False	-
SEMT.S.C03.Tx	Request Fast Poll Mode Response server command	False	-
SEMT.S.C04.Tx	`ScheduleSnapshotResponse` server command	False	-
SEMT.S.C05.Tx	`TakeSnapshotResponse` server command	False	-
SEMT.S.C06.Tx	Publish Snapshot server command	False	-
SEMT.S.C07.Tx	`GetSampledData` Response server command	True	-
SEMT.S.C08.Tx	`ConfigureMirror` server command	False	-
SEMT.S.C09.Tx	`ConfigureNotification` Scheme server command	False	-
SEMT.S.C0a.Tx	`ConfigureNotification` Flag server command	False	-
SEMT.S.C0b.Tx	`GetNotifiedMessage` server command	False	-
SEMT.S.C0c.Tx	Supply Status Response server command	False	-
SEMT.S.C0d.Tx	`StartSamplingResponse` server command	False	-
SEMT.C.Afffd	`ClusterRevision` client attribute	True	-
SEMT.C.Afffe	`AttributeReportingStatus` client attribute	False	-
SEMT.C.C00.Rsp	Get Profile Response client command	False	-
SEMT.C.C01.Rsp	Request Mirror client command	False	-

PICS Code	Name	Supported	Notes
SEMT.C.C02.Rsp	Remove Mirror client command	False	-
SEMT.C.C03.Rsp	Request Fast Poll Mode Response client command	False	-
SEMT.C.C04.Rsp	`ScheduleSnapshot` Response client command	False	-
SEMT.C.C05.Rsp	`TakeSnapshotResponse` client command	False	-
SEMT.C.C06.Rsp	Publish Snapshot client command	False	-
SEMT.C.C07.Rsp	`GetSampledData` Response client command	False	-
SEMT.C.C08.Rsp	`ConfigureMirror` client command	False	-
SEMT.C.C09.Rsp	`ConfigureNotification` Scheme client command	False	-
SEMT.C.C0a.Rsp	`ConfigureNotification` Flag client command	False	-
SEMT.C.C0b.Rsp	`GetNotifiedMessage` client command	False	-
SEMT.C.C0c.Rsp	Supply Status Response client command	False	-
SEMT.C.C0d.Rsp	`StartSamplingResponse` client command	False	-
SEMT.C.C00.Tx	Get Profile client command	True	-
SEMT.C.C01.Tx	Request Mirror Response client command	False	-
SEMT.C.C02.Tx	Mirror Removed client command	False	-
SEMT.C.C03.Tx	Request Fast Poll Mode client command	False	-
SEMT.C.C04.Tx	`ScheduleSnapshot` client command	False	-
SEMT.C.C05.Tx	`TakeSnapshot` client command	False	-
SEMT.C.C06.Tx	`GetSnapshot` client command	False	-
SEMT.C.C07.Tx	`StartSampling` client command	False	-
SEMT.C.C08.Tx	`GetSampledData` client command	True	-
SEMT.C.C09.Tx	`MirrorReportAttributeResponse` client command	False	-
SEMT.C.C0a.Tx	`ResetLoadLimit` Counter client command	False	-
SEMT.C.C0b.Tx	Change Supply client command	False	-
SEMT.C.C0c.Tx	Local Change Supply client command	True	-
SEMT.C.C0d.Tx	`SetSupplyStatus` client command	False	-
SEMT.C.C0e.Tx	`SetUncontrolledFlowThreshold` client command	False	-

5.23.2 Metering cluster functions

5.23.2.1 *ZbZclMeterClientAlloc*

```
struct ZbZclClusterT * ZbZclMeterClientAlloc(struct ZigBeeT *zb, uint8_t endpoint, struct
ZbZclMeterClientCallbacksT *callbacks, void *arg);
```

Create a new instance of the Metering Client cluster

Table 389. **ZbZclMeterClientAlloc parameters**

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster
`callbacks`	Structure containing any callback function pointers for this cluster
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: Cluster pointer, or NULL if there is an error.

5.23.2.2 *ZbZclMeterClientCommandGetProfileReq*

```
uint8_t ZbZclMeterClientCommandGetProfileReq(struct ZbZclClusterT *cluster, const struct
ZbApsAddrT *dst, struct ZbZclMeterClientGetProfileReqT *cmd_req, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Get Profile command

Table 390. **ZbZclMeterClientCommandGetProfileReq parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`cmd_req`	Get Profile command structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.23.2.3 *ZbZclMeterClientCommandGetSampledDataReq*

```
uint8_t ZbZclMeterClientCommandGetSampledDataReq(struct ZbZclClusterT *cluster, const struct
ZbApsAddrT *dst, struct ZbZclMeterClientGetSampledDataReqT *cmd_req, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Get Sampled Data command

Table 391. **ZbZclMeterClientCommandGetSampledDataReq parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`cmd_req`	Get Scheduled Events command request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.23.2.4 *ZbZclMeterClientCommandLocalChangeSupplyReq*

```
uint8_t ZbZclMeterClientCommandLocalChangeSupplyReq(struct ZbZclClusterT *cluster, const
struct ZbApsAddrT *dst, struct ZbZclMeterClientLocalChangeSupplyReqT*cmd_req, void
(*callback)(struct ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Local Change Supply command

Table 392. **ZbZclMeterClientCommandLocalChangeSupplyReq parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`cmd_req`	Get Profile command structure

Parameter	Description
callback	Callback function that is invoked when the response is received
arg	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other ZclStatusCodeT value on error.

5.23.2.5 *ZbZclMeterFormSampledData*

```
int ZbZclMeterFormSampledData(uint8_t *sample_data, unsigned int max_len, uint32_t*samples,
uint16_t num_samples);
```

Convert an array of 24-bit integers to the Zigbee frame format

Table 393. ZbZclMeterFormSampledData parameters

Parameter	Description
sample_data	Converted data for Zigbee frame
max_len	Maximum length
samples	Array of 24-bit integers to convert
num_samples	Number of integers to convert

Return: Returns the number of octets written to sample_data or -1 on error

5.23.2.6 *ZbZclMeterGetProfileIntervalPeriod*

```
int ZbZclMeterGetProfileIntervalPeriod(uint8_t profile_interval_id);
```

Convert the profile interval period enumerated value to a time in seconds

Table 394. ZbZclMeterGetProfileIntervalPeriod parameters

Parameter	Description
profile_interval_id	Zigbee® stack instance

Return: Converted time in seconds or -1 on error

5.23.2.7 *ZbZclMeterServerAlloc*

```
struct ZbZclClusterT * ZbZclMeterServerAlloc(struct ZigBeeT *zb, uint8_t endpoint, struct
ZbZclMeterServerCallbacksT *callbacks, void *arg);
```

Create a new instance of the Metering Server cluster After initialization, ZbZclClusterSetCallbackArg is called with 'arg' to configure the application callback argument

Table 395. ZbZclMeterServerAlloc parameters

Parameter	Description
zb	Zigbee® stack instance
endpoint	Endpoint on which to create cluster
callbacks	Structure containing any callback function pointers for this cluster
arg	Pointer to application data that is provided back to the callback function when invoked

Return: Cluster pointer, or NULL if there is an error.

5.23.2.8 *ZbZclMeterServerMirrorAlloc*

```
struct ZbZclClusterT * ZbZclMeterServerMirrorAlloc(struct ZigBeeT *zb, uint8_t endpoint,
uint64_t mirrorAddr, uint8_t mirrorEndpt);
```

Create a new instance of the Metering Server Mirror cluster

Table 396. **ZbZclMeterServerMirrorAlloc parameters**

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster
`mirrorAddr`	Address of the server mirror
`mirrorEndpoint`	Endpoint on which to create mirror cluster

Return: Cluster pointer, or NULL if there is an error.

5.23.2.9 *ZbZclMeterServerMirrorConfig*

```
void ZbZclMeterServerMirrorConfig(struct ZbZclClusterT *cluster, uint64_t srcExtAddr,
uint8_t dstEndpoint);
```

Configure an endpoint to mirror to a destination endpoint

Table 397. **ZbZclMeterServerMirrorConfig parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`srcExtAddr`	Source address from where reports originate
`dstEndpoint`	Destination endpoint to where the report is mirrored

Return: Void

5.23.2.10 *ZbZclMeterServerSendGetProfileRsp*

```
enum ZclStatusCodeT ZbZclMeterServerSendGetProfileRsp(struct ZbZclClusterT*cluster, struct
ZbZclAddrInfoT *dst, struct ZbZclMeterServerGetProfileRspT *rsp);
```

Send a Get Profile Response command

Table 398. **ZbZclMeterServerSendGetProfileRsp parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`rsp`	Get profile response command structure

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.23.2.11 *ZbZclMeterServerSendGetSampledDataRsp*

```
enum ZclStatusCodeT ZbZclMeterServerSendGetSampledDataRsp(struct ZbZclClusterT*cluster,
struct ZbZclAddrInfoT *dst, struct ZbZclMeterServerGetSampledDataRspT*rsp);
```

Send a Get Sampled Data Response command

Table 399. ZbZclMeterServerSendGetSampledDataRsp parameters

Parameter	Description
cluster	Cluster instance from which to send this command
dst	Requested destination address
rsp	Get sampled data response command structure

Return: ZCL_STATUS_SUCCESS if successful, or other ZclStatusCodeT value on error.

5.23.3 Metering cluster enumerations

5.23.3.1 *ZbZclMeteringSvrAttrT*

Table 400. Metering server attribute IDs

Parameter	Description
ZCL_METER_SVR_ATTR_CURSUM_DELIV	Current Summation Delivered
ZCL_METER_SVR_ATTR_CURSUM_RECV	Current Summation Received (optional)
ZCL_METER_SVR_ATTR_MAX_DMND_DELIV	Current Max Demand Delivered (optional)
ZCL_METER_SVR_ATTR_MAX_DMND_RECV	Current Max Demand Received (optional)
ZCL_METER_SVR_ATTR_DFT_SUM	DFT Summation (optional)
ZCL_METER_SVR_ATTR_DAILY_FREEZE_TIME	Daily Freeze Time (optional)
ZCL_METER_SVR_ATTR_POWER_FACTOR	Power Factor (optional)
ZCL_METER_SVR_ATTR_READ_SNAPSHOT_TIME	Reading Snap Shot Time (optional)
ZCL_METER_SVR_ATTR_MAX_DMND_DELIV_TIME	Current Max Demand Delivered Time (optional)
ZCL_METER_SVR_ATTR_MAX_DMND_RECV_TIME	Current Max Demand Received Time (optional)
ZCL_METER_SVR_ATTR_DEFAULT_UPDATE_PERIOD	Default Update Period (optional)
ZCL_METER_SVR_ATTR_FAST_POLL_UPDATE_PERIOD	Fast Poll Update Period (optional)
ZCL_METER_SVR_ATTR_BLOCK_CONSUMP_DELIV	Current Block Period Consumption Delivered (optional)
ZCL_METER_SVR_ATTR_DAILY_CONSUMPTION_TARGET	Daily Consumption Target (optional)
ZCL_METER_SVR_ATTR_CURBLOCK	Current Block (optional)
ZCL_METER_SVR_ATTR_PROFILE_INTERVAL_PERIOD	Profile Interval Period (optional)
ZCL_METER_SVR_ATTR_INTERVAL_REPORTING_PERIOD	(Deprecated) (optional)
ZCL_METER_SVR_ATTR_PRESET_READING_TIME	Preset Reading Time (optional)
ZCL_METER_SVR_ATTR_VOLUME_PER_REPORT	Summation Delivered Per Report (optional)
ZCL_METER_SVR_ATTR_FLOW_RESTRICTION	Flow Restriction (optional)
ZCL_METER_SVR_ATTR_SUPPLY_STATUS	Supply Status (optional)
ZCL_METER_SVR_ATTR_INLET_EC_SUMMATION	Current Inlet Energy Carrier Summation (optional)
ZCL_METER_SVR_ATTR_OUTLET_EC_SUMMATION	Current Outlet Energy Carrier Summation (optional)
ZCL_METER_SVR_ATTR_INLET_TEMPERATURE	Inlet Temperature (optional)
ZCL_METER_SVR_ATTR_OUTLET_TEMPERATURE	Outlet Temperature (optional)
ZCL_METER_SVR_ATTR_CONTROL_TEMPERATURE	Control Temperature (optional)

Parameter	Description
ZCL_METER_SVR_ATTR_INLET_EC_DEMAND	Current Inlet Energy Carrier Demand (optional)
ZCL_METER_SVR_ATTR_OUTLET_EC_DEMAND	Current Outlet Energy Carrier Demand (optional)
ZCL_METER_SVR_ATTR_PREVIOUS_BLOCK_DELIV	Previous Block Period Consumption Delivered (optional)
ZCL_METER_SVR_ATTR_CURBLOCK_CONSUMPTION_RECIEVED	Current Block Period Consumption Received (optional)
ZCL_METER_SVR_ATTR_CURBLOCK_RECV	Current Block Received (optional)
ZCL_METER_SVR_ATTR_DFT_SUM_RECV	DFT Summation Received (optional)
ZCL_METER_SVR_ATTR_ACTREGTIER_DELIV	Active Register Tier Delivered (optional)
ZCL_METER_SVR_ATTR_ACTREGTIER_RECV	Active Register Tier Received (optional)
ZCL_METER_SVR_ATTR_LAST_BLOCK_SWITCH_TIME	Last Block Switch Time (optional)
ZCL_METER_SVR_ATTR_TIER1_DELIV	Current Tier 1 Summation Delivered (optional) For all the tiers, use the ZCL_METER_SVR_ATTR_TIERN_DELIV(tier) macro.
ZCL_METER_SVR_ATTR_TIER1_RECV	Current Tier 1 Summation Received (optional) For all the tiers, use the ZCL_METER_SVR_ATTR_TIERN_RECV(tier) macro.
ZCL_METER_SVR_ATTR_METER_STATUS	Status
ZCL_METER_SVR_ATTR_RMN_BATT_LIFE	Remaining Battery Life (optional)
ZCL_METER_SVR_ATTR_HOURS_IN_OPERATION	Hours In Operation (optional)
ZCL_METER_SVR_ATTR_HOURS_IN_FAULT	Hours In Fault (optional)
ZCL_METER_SVR_ATTR_EXTENDED_STATUS	Extended Status (optional)
ZCL_METER_SVR_ATTR_RMN_BATT_DAYS	Remaining Battery Life in Days (optional)
ZCL_METER_SVR_ATTR_CURMETER_ID	Current Meter ID (optional)
ZCL_METER_SVR_ATTR_AMBIENT_CONSUMPTION_INDICATOR	Ambient Consumption Indicator (optional)
ZCL_METER_SVR_ATTR_UNIT_OF_MEASURE	Unit of Measure
ZCL_METER_SVR_ATTR_MULTIPLIER	Multiplier (optional)
ZCL_METER_SVR_ATTR_DIVISOR	Divisor (optional)
ZCL_METER_SVR_ATTR_SUMMATION_FORMAT	Summation Formatting
ZCL_METER_SVR_ATTR_DEMAND_FORMAT	Demand Formatting (optional)
ZCL_METER_SVR_ATTR_HISTORICAL_CONSUMPTION_FORMAT	Historical Consumption Formatting (optional)
ZCL_METER_SVR_ATTR_DEVICE_TYPE	Metering Device Type
ZCL_METER_SVR_ATTR_SITE_ID	Site ID (optional)
ZCL_METER_SVR_ATTR_SERIAL_NUMBER	Meter Serial Number (optional)
ZCL_METER_SVR_ATTR_EC_UNIT_OF_MEASURE	Energy Carrier Unit Of Measure (optional)
ZCL_METER_SVR_ATTR_EC_SUMMATION_FORMAT	Energy Carrier Summation Formatting (optional)
ZCL_METER_SVR_ATTR_EC_DEMAND_FORMAT	Energy Carrier Demand Formatting (optional)
ZCL_METER_SVR_ATTR_TEMP_UNIT_OF_MEASURE	Temperature Unit Of Measure (optional)
ZCL_METER_SVR_ATTR_TEMP_FORMAT	Temperature Formatting (optional)
ZCL_METER_SVR_ATTR_MODULE_SERIAL_NUMBER	Module Serial Number (optional)
ZCL_METER_SVR_ATTR_OPERTRFLBL_DELIV	Operating Tariff Label Delivered (optional)
ZCL_METER_SVR_ATTR_OPERTRFLBL_RECIEVED	Operating Tariff Label Received (optional)
ZCL_METER_SVR_ATTR_CUSTOMER_ID_NUMBER	Customer ID Number (optional)
ZCL_METER_SVR_ATTR_ALT_UNIT_OF_MEASURE	Alternative Unit of Measure (optional)

Parameter	Description
ZCL_METER_SVR_ATTR_ALT_DEMAND_FORMATTING	Alternative Demand Formatting (optional)
ZCL_METER_SVR_ATTR_ALT_CONSUMPTION_FORMATTING	Alternative Demand Formatting (optional)
ZCL_METER_SVR_ATTR_INSTANTANEOUS_DEMAND	Instantaneous Demand (optional)
ZCL_METER_SVR_ATTR_CURDAY_DM_DELIV	Current Day Consumption Delivered (optional)
ZCL_METER_SVR_ATTR_CURDAY_DM_RECV	Current Day Consumption Received (optional)
ZCL_METER_SVR_ATTR_PRVDAY_DM_DELIV	Previous Day Consumption Delivered (optional)
ZCL_METER_SVR_ATTR_PRVDAY_DM_RECV	Previous Day Consumption Received (optional)
ZCL_METER_SVR_ATTR_CURPRTL_DELIV_START	Current Partial Profile Interval Start Time Delivered (optional)
ZCL_METER_SVR_ATTR_CURPRTL_RECV_START	Current Partial Profile Interval Start Time Received (optional)
ZCL_METER_SVR_ATTR_CURPRTL_DELIV	Current Partial Profile Interval Value Delivered (optional)
ZCL_METER_SVR_ATTR_CURPRTL_RECV	Current Partial Profile Interval Value Received (optional)
ZCL_METER_SVR_ATTR_CURDAY_MAX_PRESSURE	Current Day Max Pressure (optional)
ZCL_METER_SVR_ATTR_CURDAY_MIN_PRESSURE	Current Day Min Pressure (optional)
ZCL_METER_SVR_ATTR_PRVDAY_MAX_PRESSURE	Previous Day Max Pressure (optional)
ZCL_METER_SVR_ATTR_PRVDAY_MIN_PRESSURE	Previous Day Min Pressure (optional)
ZCL_METER_SVR_ATTR_CURDAY_MAX_DEMAND	Current Day Max Demand (optional)
ZCL_METER_SVR_ATTR_PRVDAY_MAX_DEMAND	Previous Day Max Demand (optional)
ZCL_METER_SVR_ATTR_CURMTH_MAX_DEMAND	`Current Month Max Demand` (optional)
ZCL_METER_SVR_ATTR_CURYR_MAX_DEMAND	Current Year Max Demand (optional)
ZCL_METER_SVR_ATTR_CURDAY_EC_MAX_DEMAND	Current Day Max Energy Carrier Demand (optional)
ZCL_METER_SVR_ATTR_PRVDAY_EC_MAX_DEMAND	Previous Day Max Energy Carrier Demand (optional)
ZCL_METER_SVR_ATTR_CURMTH_EC_MAX_DEMAND	Current Month Max Energy Carrier Demand (optional)
ZCL_METER_SVR_ATTR_CURMTH_EC_MIN_DEMAND	Current Month Min Energy Carrier Demand (optional)
ZCL_METER_SVR_ATTR_CURYR_EC_MAX_DEMAND	Current Year Max Energy Carrier Demand (optional)
ZCL_METER_SVR_ATTR_CURYR_EC_MIN_DEMAND	Current Year Min Energy Carrier Demand (optional)
ZCL_METER_SVR_ATTR_PRVDAY_2_DELIV_CONSUMP	Previous Day 2 Consumption Delivered (optional) ZCL_METER_SVR_ATTR_PRVDAY_N_DELIV_CONSUMP(2) For all the previous days, use the ZCL_METER_SVR_ATTR_PRVDAY_N_DELIV_CONSUMP(prev_day) macro.
ZCL_METER_SVR_ATTR_PRVDAY_2_RECV_CONSUMP	Previous Day 2 Consumption Received (optional) ZCL_METER_SVR_ATTR_PRVDAY_N_RECV_CONSUMP(2) For all the previous days, use the ZCL_METER_SVR_ATTR_PRVDAY_N_RECV_CONSUMP(prev_day) macro.
ZCL_METER_SVR_ATTR_WEEK_0_CONSUMP_DELIV	Current Week Consumption Delivered (optional) - ZCL_METER_SVR_ATTR_WEEK_N_CONSUMP_DELI V(0)
ZCL_METER_SVR_ATTR_WEEK_0_CONSUMP_RECV	Current Week Consumption Received (optional) - ZCL_METER_SVR_ATTR_WEEK_N_CONSUMP_REC V(0)
ZCL_METER_SVR_ATTR_WEEK_1_CONSUMP_DELIV	Previous Week Consumption Delivered (optional) - ZCL_METER_SVR_ATTR_WEEK_N_CONSUMP_DELI V(1)

Parameter	Description
ZCL_METER_SVR_ATTR_WEEK_1_CONSUMP_RECV	Previous Week Consumption Received (optional) - ZCL_METER_SVR_ATTR_WEEK_N_CONSUMP_REC V(1)
ZCL_METER_SVR_ATTR_WEEK_2_CONSUMP_DELIV	Previous Week 2 Consumption Delivered (optional) ZCL_METER_SVR_ATTR_WEEK_N_CONSUMP_DELIV(2) For all the week consumption delivered, use the ZCL_METER_SVR_ATTR_WEEK_N_CONSUMP_DELIV(prev_week) macro.
ZCL_METER_SVR_ATTR_WEEK_2_CONSUMP_RECV	Previous Week 2 Consumption Received (optional) ZCL_METER_SVR_ATTR_WEEK_N_CONSUMP_RECV(2) For all the week consumption received, use the ZCL_METER_SVR_ATTR_WEEK_N_CONSUMP_RECV(prev_week) macro.
ZCL_METER_SVR_ATTR_MONTH_0_CONSUMP_DEL IV	Current Month Consumption Delivered (optional) - ZCL_METER_SVR_ATTR_MONTH_N_CONSUMP_DEL IV(0)
ZCL_METER_SVR_ATTR_MONTH_0_CONSUMP_REC V	Current Month Consumption Received (optional) - ZCL_METER_SVR_ATTR_MONTH_N_CONSUMP_RE CV(0)
ZCL_METER_SVR_ATTR_MONTH_1_CONSUMP_DEL IV	Previous Month Consumption Delivered (optional) - ZCL_METER_SVR_ATTR_MONTH_N_CONSUMP_DEL IV(1)
ZCL_METER_SVR_ATTR_MONTH_1_CONSUMP_REC V	Previous Month Consumption Received (optional) - ZCL_METER_SVR_ATTR_MONTH_N_CONSUMP_RE CV(1)
ZCL_METER_SVR_ATTR_MONTH_2_CONSUMP_DEL IV	Previous Month 2 Consumption Delivered (optional) ZCL_METER_SVR_ATTR_MONTH_N_CONSUMP_DELIV(2) For all the month consumption delivered, use the ZCL_METER_SVR_ATTR_MONTH_N_CONSUMP_DELIV(prev_month) macro.
ZCL_METER_SVR_ATTR_MONTH_2_CONSUMP_REC V	Previous Month 2 Consumption Received (optional) ZCL_METER_SVR_ATTR_MONTH_N_CONSUMP_RECV(2) For all the prev_month consumption received, use the ZCL_METER_SVR_ATTR_MONTH_N_CONSUMP_RECV(prev_month) macro.
ZCL_METER_SVR_ATTR_HISTORICAL_FREEZE_TIM E	Historical Freeze Time (optional)
ZCL_METER_SVR_ATTR_MAX_PERIODS_DELIV	Max Number Of Periods Delivered (optional)
ZCL_METER_SVR_ATTR_CURDEMAND_DELIV	Current Demand Delivered (optional)
ZCL_METER_SVR_ATTR_DEMAND_LIMIT	Demand Limit (optional)
ZCL_METER_SVR_ATTR_DEMAND_INTEGRATION_P ERIOD	Demand Integration Period (optional)
ZCL_METER_SVR_ATTR_DEMAND_SUBINTERVALS	Number Of Demand Subintervals (optional)
ZCL_METER_SVR_ATTR_SUPPLY_LIMIT_ARM_DUR ATION	Demand Limit Arm® Duration (optional)
ZCL_METER_SVR_ATTR_LOAD_LIMIT_SUPPLY_STA TE	Load Limit Supply State (optional)
ZCL_METER_SVR_ATTR_LOAD_LIMIT_COUNTER	Load Limit Counter (optional)
ZCL_METER_SVR_ATTR_SUPPLY_TAMPER_STATE	Supply Tamper State (optional)
ZCL_METER_SVR_ATTR_SUPPLY_DEPLETION_STA TE	Supply Depletion State (optional)
ZCL_METER_SVR_ATTR_SUPPLY_UNCONTROLLED _FLOW_STATE	Supply Uncontrolled Flow State (optional)
ZCL_METER_SVR_ATTR_NO_TIER_BLOCK1_DELIV	Current No Tier Block 1 Summation Delivered (optional) ZCL_METER_SVR_ATTR_NO_TIER_BLOCKN_DELIV(1) For all Current No Tier Block N Summation Delivered, use the ZCL_METER_SVR_ATTR_NO_TIER_BLOCKN_DELIV(block) macro.

Parameter	Description
ZCL_METER_SVR_ATTR_TIER1_BLOCK1_DELIV	Current Tier 1 Block 1 Summation Delivered (optional) ZCL_METER_SVR_ATTR_TIERN_BLOCKN_DELIV(1,1). For all Current Tier N Block N Summation Delivered, use the ZCL_METER_SVR_ATTR_TIERN_BLOCKN_DELIV(tier, block) macro.
ZCL_METER_SVR_ATTR_GENERIC_ALARM_MASK	Generic Alarm Mask
ZCL_METER_SVR_ATTR_ELECTRICITY_ALARM_MASK	Electricity Alarm Mask
ZCL_METER_SVR_ATTR_FLOW_PRESSURE_ALARM_MASK	Generic Flow/Pressure Alarm Mask
ZCL_METER_SVR_ATTR_WATER_ALARM_MASK	Water Specific Alarm Mask
ZCL_METER_SVR_ATTR_HEAT_COOLING_ALARM_MASK	Heat and Cooling Specific Alarm Mask
ZCL_METER_SVR_ATTR_GAS_ALARM_MASK	Gas Specific `AlarmMask`
ZCL_METER_SVR_ATTR_EXTENDED_GENERIC_ALARM_MASK	Extended Generic AlarmMask
ZCL_METER_SVR_ATTR_MANUFACTURER_ALARM_MASK	Manufacturer AlarmMask
ZCL_METER_SVR_ATTR_NO_TIER_BLOCK1_RECV	Current No Tier Block 1 Summation Received (optional) ZCL_METER_SVR_ATTR_NO_TIER_BLOCKN_RECV(1) For all Current No Tier Block N Summation Received, use the ZCL_METER_SVR_ATTR_NO_TIER_BLOCKN_RECV(block) macro.
ZCL_METER_SVR_ATTR_TIER1_BLOCK1_RECV	Current Tier 1 Block 1 Summation Received (optional) ZCL_METER_SVR_ATTR_TIERN_BLOCKN_RECV(1,1) For all Current Tier N Block N Summation Received, use the ZCL_METER_SVR_ATTR_TIERN_BLOCKN_RECV(tier, block) macro.
ZCL_METER_SVR_ATTR_BTD_DELIV	Bill To Date Delivered (optional)
ZCL_METER_SVR_ATTR_BTD_TIMESTAMP_DELIV	Bill To Date Time Stamp Delivered (optional)
ZCL_METER_SVR_ATTR_PROJ_BILL_DELIV	Projected Bill Delivered (optional)
ZCL_METER_SVR_ATTR_PROJ_BILL_TIMESTAMP_DELIV	Projected Bill Time Stamp Delivered (optional)
ZCL_METER_SVR_ATTR_BILL_DELIV_TRAILING_DIGIT	Bill Delivered Trailing Digit (optional)
ZCL_METER_SVR_ATTR_BTD_RECV	Bill To Date Received (optional)
ZCL_METER_SVR_ATTR_BTD_TIMESTAMP_RECV	Bill To Date Time Stamp Received (optional)
ZCL_METER_SVR_ATTR_PROJ_BILL_RECV	Projected Bill Received (optional)
ZCL_METER_SVR_ATTR_PROJ_BILL_TIMESTAMP_RECV	Projected Bill Time Stamp Received (optional)
ZCL_METER_SVR_ATTR_BILL_RECV_TRAILING_DIGIT	Projected Bill Time Stamp Received (optional)
ZCL_METER_SVR_ATTR_PROP_CHG_SUPPLY_IMPL_TIME	Proposed Change Supply Implementation Time (optional)
ZCL_METER_SVR_ATTR_PROP_CHG_SUPPLY_STATUS	Proposed Change Supply Status (optional)
ZCL_METER_SVR_ATTR_UNCTLFLOW_THRESHOLD	Uncontrolled Flow Threshold (optional)
ZCL_METER_SVR_ATTR_UNCTLFLOW_UNIT_OF_MEASURE	Uncontrolled Flow Threshold Unit of Measure (optional)
ZCL_METER_SVR_ATTR_UNCTLFLOW_MULTIPLIER	Uncontrolled Flow Multiplier (optional)

Parameter	Description
ZCL_METER_SVR_ATTR_UNCTLFLOW_DIVISOR	Uncontrolled Flow Divisor (optional)
ZCL_METER_SVR_ATTR_FLOW_STABILISATION_PERIOD	Flow Stabilisation Period (optional)
ZCL_METER_SVR_ATTR_FLOW_MEASUREMENT_PERIOD	Flow Measurement Period (optional)
ZCL_METER_SVR_ATTR_ALT_INSTANTANEOUS_DEMAND	Alternative Instantaneous Demand (optional)

5.23.4 Metering cluster structures

5.23.4.1 *ZbZclMeterClientCallbacksT*

Metering Client callbacks configuration

Table 401. ZbZclMeterClientCallbacksT parameters

Parameter	Description
`optional` (callback function pointer)	`Enum ZclStatusCodeT(*optional)(struct ZbZclClusterT *cluster,struct ZbZclHeaderT *zclHdrPtr,struct ZbApsdeDataIndT *dataIndPtr)` Optional commands not parsed into a struct or specific callback. If this is NULL, the cluster returns a with status of ZCL_STATUS_UNSUPP_CLUSTER_COMMAND. ZCL_STATUS_SUCCESS if successful, or other value on error.

5.23.4.2 *ZbZclMeterClientGetProfileReqT*

`ZclStatusCodeT`

Get Profile command structure

Table 402. ZbZclMeterClientGetProfileReqT parameters

Parameter	Description
`enum ZbZclMeterSampleTypeTinterval_channel`	Interval channel
`uint32_t end_time`	End time
`uint8_t number_of_periods`	Number of periods

5.23.4.3 *ZbZclMeterClientGetSampledDataReqT*

Get Sampled Data command structure

Table 403. ZbZclMeterClientGetSampledDataReqT parameters

Parameter	Description
`uint16_t sample_id`	Sample ID
`uint32_t earliest_sample_time`	Earliest sample time
`enum ZbZclMeterSampleTypeTsample_type`	Sample type
`uint16_t number_of_samples`	Number of samples

5.23.4.4 *ZbZclMeterClientLocalChangeSupplyReqT*

Metering Local Change Supply command structure

Table 404. ZbZclMeterClientLocalChangeSupplyReqT parameters

Parameter	Description
`enum ZbZclMeterSupplyStatusTprop_supply_status`	Proposed Supply Status

5.23.4.5 *ZbZclMeterServerCallbacksT*

Metering Server callbacks configuration

Table 405. ZbZclMeterServerCallbacksT parameters

Parameter	Description
`get_profile` (callback function pointer)	`enum ZclStatusCodeT (*get_profile)(struct ZbZclClusterT *cluster, void *arg,struct ZbZclMeterClientGetProfileReqT *req, struct ZbZclAddrInfoT *srcInfo)` Configure callback to handle ZCL_METER_CLI_CMD_GET_PROFILE. In handling of `get_profile()`, the application should call `ZbZclMeterServerSendGetProfileRsp` to send the response if successful. ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.
`get_sampled_data` (callback function pointer)	`enum ZclStatusCodeT (*get_sampled_data)(struct ZbZclClusterT *cluster, void *arg,struct ZbZclMeterClientGetSampledDataReqT *req, struct ZbZclAddrInfoT *srcInfo)` Configure callback to handle ZCL_METER_CLI_CMD_GET_SAMPLED_DATA. In handling of `get_sampled_data()`, the application should call `ZbZclMeterServerSendGetSampledDataRsp` to send the response if successful. ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.
`local_change_supply` (callback function pointer)	`enum ZclStatusCodeT (*local_change_supply)(struct ZbZclClusterT *cluster, void *arg,struct ZbZclMeterClientLocalChangeSupplyReqT *req, struct ZbZclAddrInfoT *srcInfo)` Configure callback to handle ZCL_METER_CLI_CMD_LOCAL_CHANGE_SUPPLY. ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.
`request_mirror` (callback function pointer)	`uint16_t (*request_mirror)(void *arg, struct ZbZclClusterT *cluster,struct ZbZclAddrInfoT *srcInfo, struct ZbApsdeDataIndT *dataIndPtr)` Configure callback to handle ZCL_METER_CLI_CMD_REQUEST_MIRROR_RESPONSE command. Callback registers an endpoint for the mirrored metering cluster, and return the endpoint ID or 0xFFFF on error.
`remove_mirror` (callback function pointer)	`uint16_t (*remove_mirror)(void *arg, struct ZbZclClusterT *cluster,struct ZbZclAddrInfoT *srcInfo, struct ZbApsdeDataIndT *dataIndPtr)` Configure callback to handle ZCL_METER_CLI_CMD_MIRROR_REMOVED command. Callback registers an endpoint for the mirrored metering cluster, and return the endpoint ID or 0xFFFF on error.
`optional` (callback function pointer)	`enum ZclStatusCodeT (*optional)(struct ZbZclClusterT *cluster,struct ZbZclHeaderT *zclHdrPtr, struct ZbApsdeDataIndT *dataIndPtr)` Optional commands not parsed into a struct or specific callback. If this is NULL, the cluster returns a default response with the status of ZCL_STATUS_UNSUPP_CLUSTER_COMMAND. ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.23.4.6 *ZbZclMeterServerGetProfileRspT*

Get Profile Response command structure

Table 406. ZbZclMeterServerGetProfileRspT parameters

Parameter	Description
`uint32_t end_time`	End Time
`enum ZbZclMeterGetProfileStatusTstatus`	Status
`uint8_t profile_interval_period`	Profile interval period
`uint8_t number_of_periods`	Number of periods delivered
`uint8_t *profile_data`	Profile buffer
`unsigned int profile_length`	Profile buffer length

5.23.4.7 *ZbZclMeterServerGetSampledDataRspT*

Get Sampled Data Response command structure

Table 407. ZbZclMeterServerGetSampledDataRspT parameters

Parameter	Description
`uint16_t sample_id`	Sample ID
`uint32_t sample_start_time`	Sample start time
`enum ZbZclMeterSampleTypeT sample_type`	Sample type
`uint16_t sample_request_interval`	Sample request interval
`uint16_t number_of_samples`	Number of samples
`uint8_t *sample_data`	Sampled data buffer
`unsigned int sample_length`	Sampled data buffer length

5.24 Meter identification cluster

```
#include "zcl/general/zcl.meter.id.h"
```

5.24.1 Meter identification cluster description

Table 408. Meter identification cluster PICS code description

PICS Code	Name	Supported	Notes
MTRID.S	Server	True	-
MTRID.C	Client	True	-
MTRID.S.A0000	`CompanyName` server attribute	True	-
MTRID.S.A0001	`MeterTypeID` server attribute	True	-
MTRID.S.A0004	`DataQualityID` server attribute	True	-
MTRID.S.A000c	`POD` server attribute	True	-
MTRID.S.A000d	`AvailablePower` server attribute	True	-
MTRID.S.A000e	`PowerThreshold` server attribute	True	-
MTRID.S.Afffd	`ClusterRevision` server attribute	True	-
MTRID.S.Afffe	`AttributeReportingStatus` server attribute	False	-

5.24.2 Meter identification cluster functions

5.24.2.1 *ZbZclMeterIdClientAlloc*

```
struct ZbZclClusterT * ZbZclMeterIdClientAlloc(struct ZigBeeT *zb, uint8_t endpoint);
```

Instantiate a new instance of the Meter Identification Client cluster

Table 409. ZbZclMeterIdClientAlloc parameters

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster

Return: Cluster pointer, or NULL if there is an error.

5.24.2.2 *ZbZclMeterIdServerAlloc*

```
struct ZbZclClusterT * ZbZclMeterIdServerAlloc(struct ZigBeeT *zb, uint8_t endpoint);
```

Instantiate a new instance of the Meter Identification Server cluster

Table 410. ZbZclMeterIdServerAlloc parameters

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster

Return: Cluster pointer, or NULL if there is an error.

5.24.3 Meter identification cluster enumerations

5.24.3.1 *ZbZclMeterIdSvrAttrT*

Table 411. ZbZclMeterIdSvrAttrT attribute identifiers

Attributes	Description
ZCL_METER_ID_ATTR_COMPANY_NAME	Company name
ZCL_METER_ID_ATTR_METER_TYPE_ID	Meter type ID
ZCL_METER_ID_ATTR_DATA_QUAL_ID	Data quality ID
ZCL_METER_ID_ATTR_CUSTOMER_NAME	Customer name (optional)
ZCL_METER_ID_ATTR_MODEL	Model (optional)
ZCL_METER_ID_ATTR_PART_NUMBER	Part number (optional)
ZCL_METER_ID_ATTR_PRODUCT_REV	Product revision (optional)
ZCL_METER_ID_ATTR_SOFTWARE_REV	Software revision (optional)
ZCL_METER_ID_ATTR_UTILITY_NAME	Utility name (optional)
ZCL_METER_ID_ATTR_POD	POD
ZCL_METER_ID_ATTR_AVAILABLE_POWER	Available power
ZCL_METER_ID_ATTR_POWER_THRESH	Power threshold

5.25 Nearest gateway cluster

```
#include "zcl/general/zcl.nearest.gw.h"
```

5.25.1 Nearest gateway cluster description

Table 412. Nearest gateway cluster PISC code description

PICS Code	Name	Supported	Notes
NEARGW.S	Server	True	-
NEARGW.C	Client	True	-
NEARGW.S.A0000	Nearest Gateway server attribute	True	-
NEARGW.S.A0001	New Mobile Node server attribute	True	-
NEARGW.S.Afffd	`ClusterRevision` server attribute	True	-
NEARGW.S.Affe	`AttributeReportingStatus` server attribute	False	-
NEARGW.C.Afffd	`ClusterRevision` client attribute	True	-
NEARGW.C.Affe	`AttributeReportingStatus` client attribute	False	-

5.25.2 Nearest gateway cluster functions

5.25.2.1 *ZbZclNearestGwClientAlloc*

```
struct ZbZclClusterT * ZbZclNearestGwClientAlloc(struct ZigBeeT *zb, uint8_t endpoint);
```

Instantiate a new instance of the Nearest Gateway client cluster

Table 413. ZbZclNearestGwClientAlloc parameters

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster

Return: Cluster pointer, or NULL if there is an error.

5.25.2.2 *ZbZclNearestGwServerAlloc*

```
struct ZbZclClusterT * ZbZclNearestGwServerAlloc(struct ZigBeeT *zb, uint8_t endpoint);
```

Instantiate a new instance of the Nearest Gateway server cluster

Table 414. ZbZclNearestGwServerAlloc parameters

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster

Return: Cluster pointer, or NULL if there is an error.

5.25.3 Nearest gateway cluster enumerations

5.25.3.1 *ZbZclNearestGwServerAttrT*

Nearest Gateway Attribute Ids

Table 415. **parameters**

Parameter	Description
ZCL_NEAREST_GW_SVR_ATTR_NEAREST_GW	Nearest gateway
ZCL_NEAREST_GW_SVR_ATTR_NEW_MOBILE_NOD E	New mobile node

5.26 Occupancy sensing cluster

```
#include "zcl/general/zcl.occupancy.h"
```

5.26.1 Occupancy sensing cluster description

Table 416. **Occupancy sensing cluster PISC code description**

PICS Code	Name	Supported	Notes
OS.S	Server	True	-
OS.C	Client	True	-
OS.S.OST00	PIR	False	-
OS.S.OST01	Ultrasonic	False	-
OS.S.OST02	PIR and ultrasonic	False	-
OS.S.A0000	Occupancy server attribute	True	-
OS.S.A0000.Report.Tx	Occupancy server attribute reports	True	-
OS.S.A0001	`OccupancySensorType` server attribute	True	-
OS.S.A0002	`OccupancySensorTypeBitmap` server attribute	True	-
OS.S.A0010	`PIROccupiedToUnoccupiedDelay` server attribute	True	Optional
OS.S.A0011	`PIRUnoccupiedToOccupiedDelay` server attribute	True	Optional
OS.S.A0012	`PIRUnoccupiedToOccupiedThreshold` server attribute	True	Optional
OS.S.A0020	`UltrasonicOccupiedToUnoccupiedDelay` server attribute	True	Optional
OS.S.A0021	`UltrasonicUnoccupiedToOccupiedDelay` server attribute	True	Optional
OS.S.A0022	`UltrasonicUnoccupiedToOccupiedThreshold` server attribute	True	Optional
OS.S.Afffd	`ClusterRevision` server attribute	True	-
OS.S.Afffe	`AttributeReportingStatus` server attribute	False	-
OS.C.A0000.Report.Rsp	Occupancy client attribute reports	False	-
OS.C.Afffd	`ClusterRevision` client attribute	True	-
OS.C.Afffe	`AttributeReportingStatus` client attribute	False	-

5.26.2 Occupancy sensing cluster functions

5.26.2.1 ZbZclOccupancyClientAlloc

```
struct ZbZclClusterT * ZbZclOccupancyClientAlloc(struct ZigBeeT *zb, uint8_t endpoint);
```

Create a new instance of the Occupancy Sensing Server cluster

Table 417. **ZbZclOccupancyClientAlloc parameters**

Parameter	Description
`zb`	Zigbee® stack instance

Parameter	Description
endpoint	Endpoint on which to create cluster

Return: Cluster pointer, or NULL if there is an error.

5.26.2.2 *ZbZclOccupancyServerAlloc*

```
struct ZbZclClusterT * ZbZclOccupancyServerAlloc(struct ZigBeeT *zb, uint8_t endpoint);
```

Create a new instance of the Occupancy Sensing Client cluster

Table 418. ZbZclOccupancyServerAlloc parameters

Parameter	Description
zb	Zigbee® stack instance
endpoint	Endpoint on which to create cluster

Return: Cluster pointer, or NULL if there is an error.

5.26.3 Occupancy sensing cluster enumerations

5.26.3.1 *ZbZclOccupancySvrAttrT*

Table 419. Occupancy Sensing server attribute IDs

Parameter	Description
ZCL_OCC_ATTR_OCCUPANCY	Occupancy
ZCL_OCC_ATTR_SENSORTYPE	Occupancy sensor type
ZCL_OCC_ATTR_SENSORTYPE_BITMAP	Occupancy sensor type bitmap
ZCL_OCC_ATTR_PIR_OU_DELAY	PIR occupied to unoccupied delay (optional)
ZCL_OCC_ATTR_PIR_UO_DELAY	PIR unoccupied to occupied delay (optional)
ZCL_OCC_ATTR_PIR_UO_THRESHOLD	PIR unoccupied to occupied threshold (optional)
ZCL_OCC_ATTR_US_OU_DELAY	Ultrasonic occupied to unoccupied delay (optional)
ZCL_OCC_ATTR_US_UO_DELAY	Ultrasonic unoccupied to occupied delay (optional)
ZCL_OCC_ATTR_US_UO_THRESHOLD	Ultrasonic unoccupied to occupied threshold (optional)
ZCL_OCC_ATTR_PHY_OU_DELAY	Physical contact occupied to unoccupied delay (optional)
ZCL_OCC_ATTR_PHY_UO_DELAY	Physical contact unoccupied to occupied delay (optional)
ZCL_OCC_ATTR_PHY_UO_THRESHOLD	Physical contact unoccupied to occupied threshold (optional)

5.27 On/Off switch configuration cluster

```
#include "zcl/general/zcl.onoff.swconfig.h"
```

5.27.1 On/Off switch configuration cluster description

Table 420. On/Off switch configuration cluster PICS code description

PICS Code	Name	Supported	Notes
OOSC.S	Server	True	-
OOSC.C	Client	True	-

PICS Code	Name	Supported	Notes
OOSC.S.A0000	`SwitchType` server attribute	True	-
OOSC.S.A0010	`SwitchActions` server attribute	True	-
OOSC.S.Afffd	`ClusterRevision` server attribute	True	-
OOSC.S.Afffe	`AttributeReportingStatus` server attribute	False	-
OOSC.S.Afffd	`ClusterRevision` server attribute	True	-
OOSC.S.Afffe	`AttributeReportingStatus` server attribute	False	-

5.27.2 On/Off switch configuration cluster functions

5.27.2.1 ZbZclOnOffSwConfigClientAlloc

```
struct ZbZclClusterT * ZbZclOnOffSwConfigClientAlloc(struct ZigBeeT *zb, uint8_t endpoint);
```

Create a new instance of the OnOff Switch Configuration Client cluster

Table 421. ZbZclOnOffSwConfigClientAlloc parameters

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster

Return: Cluster pointer, or NULL if there is an error.

5.27.2.2 ZbZclOnOffSwConfigServerAlloc

```
struct ZbZclClusterT * ZbZclOnOffSwConfigServerAlloc(struct ZigBeeT *zb, uint8_t endpoint,
uint8_t switch_type);
```

Create a new instance of the OnOff Switch Configuration Server cluster

Table 422. ZbZclOnOffSwConfigServerAlloc parameters

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster
`switch_type`	Type of switch associated with this cluster server

Return: Cluster pointer, or NULL if there is an error.

5.27.3 On/Off switch configuration cluster enumerations

5.27.3.1 ZbZclOnOffSwConfigSvrAttrId

Table 423. On/off Switch Configuration cluster attribute IDs

Parameter	Description
ZCL_ONOFF_SWCONFIG_ATTR_TYPE	Switch type
ZCL_ONOFF_SWCONFIG_ATTR_ACTIONS	Switch actions

5.28 On/Off cluster

```
#include "zcl/general/zcl.onoff.h"
```

5.28.1 On/Off cluster description

Table 424. On/Off cluster PICS code description

PICS Code	Name	Supported	Notes
OO.S	Server	True	-
OO.C	Client	True	-
OO.S.A0000	`OnOff` server attribute	True	-
OO.S.A0000.Scene	`OnOff` server attribute scenes	True	-
OO.S.A0000.Report.Tx	`OnOff` server attribute reports	True	-
OO.S.Afffd	`ClusterRevision` server attribute	True	-
OO.S.C00.Rsp	Off server command	True	-
OO.S.C01.Rsp	On server command	True	-
OO.S.C02.Rsp	Toggle server command	True	-
OO.S.C40.Rsp	Off with effect server command	False	-
OO.S.C41.Rsp	On with recall global scene server command	False	-
OO.S.C42.Rsp	On with timed off server command	False	-
OO.C.A0000.Report.Rsp	`OnOff` client attribute reports	False	-
OO.C.Afffd	`ClusterRevision` client attribute	True	-
OO.C.C00.Tx	Off client command	True	-
OO.C.C01.Tx	On client command	True	-
OO.C.C02.Tx	Toggle client command	True	-
OO.C.C40.Tx	Off with effect client command	False	-
OO.C.C41.Tx	On with recall global scene client command	False	-
OO.C.C42.Tx	On with timed off client command	False	-

5.28.2 On/Off cluster functions

5.28.2.1 ZbZclOnOffClientAlloc

```
struct ZbZclClusterT * ZbZclOnOffClientAlloc(struct ZigBeeT *zb, uint8_t endpoint);
```

Create a new instance of the On/Off Client cluster

Table 425. ZbZclOnOffClientAlloc parameters

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster

Return: Cluster pointer, or NULL if there is an error.

5.28.2.2 *ZbZclOnOffClientOffReq*

```
enum ZclStatusCodeT ZbZclOnOffClientOffReq(struct ZbZclClusterT *cluster, const struct
ZbApsAddrT *dst, void (*callback)(struct ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send an Off command

Table 426. **ZbZclOnOffClientOffReq parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.28.2.3 *ZbZclOnOffClientOnReq*

```
enum ZclStatusCodeT ZbZclOnOffClientOnReq(struct ZbZclClusterT *cluster, const struct
ZbApsAddrT *dst, void (*callback)(struct ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send an On command

Table 427. **ZbZclOnOffClientOnReq parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.28.2.4 *ZbZclOnOffClientToggleReq*

```
enum ZclStatusCodeT ZbZclOnOffClientToggleReq(struct ZbZclClusterT *cluster, const struct
ZbApsAddrT *dst, void (*callback)(struct ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Toggle command

Table 428. **ZbZclOnOffClientToggleReq parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.28.2.5 *ZbZclOnOffServerAlloc*

```
struct ZbZclClusterT * ZbZclOnOffServerAlloc(struct ZigBeeT *zb, uint8_t endpoint, struct
ZbZclOnOffServerCallbacksT *callbacks, void *arg);
```

Create a new instance of the OnOff Server cluster

Table 429. ZbZclOnOffServerAlloc parameters

Parameter	Description
zb	Zigbee® stack instance
endpoint	Endpoint on which to create cluster
callbacks	Structure containing any callback function pointers for this cluster
arg	Pointer to application data that is provided back to the callback function when invoked

Return: Cluster pointer, or NULL if there is an error.

5.28.2.6 *ZbZclOnOffServerSetLevelControlCallback*

```
void ZbZclOnOffServerSetLevelControlCallback(struct ZbZclClusterT *on_off_cluster, struct
ZbZclClusterT *level_cluster, ZbZclLevelControlCallbackT levelControlCallback);
```

Set the Level control callback

Table 430. ZbZclOnOffServerSetLevelControlCallback parameters

Parameter	Description
on_off_cluster	OnOff cluster instance
level_cluster	Level cluster instance
levelControlCallback	Level control callback function

Return: Void

5.28.3 On/Off cluster enumerations

5.28.3.1 *ZbZclOnOffSvrAttrT*

Table 431. ZbZclOnOffSvrAttrT attributes

Attributes	Description
ZCL_ONOFF_ATTR_ONOFF	OnOff
ZCL_ONOFF_ATTR_GLOBAL_SCENE_CONTROL	Global scene control (optional)
ZCL_ONOFF_ATTR_ON_TIME	On time (optional)
ZCL_ONOFF_ATTR_OFF_WAIT_TIME	Off wait time (optional)

5.28.4 On/Off cluster structures

5.28.4.1 *ZbZclOnOffServerCallbacksT*

OnOff Server callbacks configuration

Table 432. ZbZclOnOffServerCallbacksT parameters

Parameter	Description
`off` (callback function pointer)	`Enum ZclStatusCodeT(*off)(struct ZbZclClusterT *cluster, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of an Off command. The application is expected to update ZCL_ONOFF_ATTR_ONOFF.
`on` (callback function pointer)	`enum ZclStatusCodeT (*on)(struct ZbZclClusterT *cluster, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of an On command. The application is expected to update ZCL_ONOFF_ATTR_ONOFF.
`toggle` (callback function pointer)	`Enum ZclStatusCodeT(*toggle)(struct ZbZclClusterT *cluster, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of a Toggle command. The application is expected to update ZCL_ONOFF_ATTR_ONOFF.

5.29 Over-The-Air upgrade cluster

```
#include "zcl/general/zcl.ota.h"
```

5.29.1 Over-The-Air upgrade cluster description

Table 433. Over-The-Air upgrade cluster PICS code description

PICS Code	Name	Supported	Notes
OUI1	See Test Spec	True	-
OUI2a	See Test Spec	False	-
OUI2b	See Test Spec	True	-
OTA.S	Server	True	-
OTA.C	Client	True	-
OTA.S.Afffd	`ClusterRevision` server attribute	True	-
OTA.S.Afffe	`AttributeReportingStatus` server attribute	False	-
OTA.S.C01.Rsp	Query Next Image Request server command	True	-
OTA.S.C03.Rsp	Image Block Request server command	True	-
OTA.S.C04.Rsp	Image Page Request server command	False	-
OTA.S.C06.Rsp	Upgrade End Request server command	True	-
OTA.S.C08.Rsp	Query Device Specific File Request server command	False	-
OTA.S.C00.Tx	Image Notify server command	True	-
OTA.S.C02.Tx	Query Next Image Response server command	True	-
OTA.S.C05.Tx	Image Block Response server command	True	-
OTA.S.C07.Tx	Upgrade End Response server command	True	-
OTA.S.C09.Tx	Query Device Specific File Response server command	False	-
OOMTS6	See Test Spec	False	-
OUPS1	See Test Spec	True	-
OUPS2	See Test Spec	True	Customer application can override
OUPS3	See Test Spec	False	-

PICS Code	Name	Supported	Notes
OUPS4	See Test Spec	True	-
OUPS5	See Test Spec	False	-
INPT01	See Test Spec	True	-
INPT02	See Test Spec	True	-
INPT03	See Test Spec	True	-
OTA.C.A0000	`UpgradeServerID` client attribute	True	-
OTA.C.A0001	`FileOffset` client attribute	True	Optional
OTA.C.A0002	`CurrentFileVersion` client attribute	True	Optional
OTA.C.A0003	`CurrentZigBeeStackVersion` client attribute	True	Optional
OTA.C.A0004	`DownloadedFileVersion` client attribute	True	Optional
OTA.C.A0005	`DownloadedZigBeeStackVersion` client attribute	True	Optional
OTA.C.A0006	`ImageUpgradeStatus` client attribute	True	Optional
OTA.C.A0007	`ManufacturerID` client attribute	True	Optional
OTA.C.A0008	`ImageTypeID` client attribute	True	Optional
OTA.C.A0009	`MinimumBlockPeriod` client attribute	True	Optional
OTA.C.A000a	`ImageStamp` client attribute	True	Optional
OTA.C.A000b	`UpgradeActivationPolicy` client attribute	True	Optional
OTA.C.A000c	`UpgradeTimeoutPolicy` client attribute	True	Optional
OTA.C.Afffd	`ClusterRevision` client attribute	True	-
OTA.C.Afffe	`AttributeReportingStatus` client attribute	False	-
OTA.C.C00.Rsp	Image Notify client command	True	-
OTA.C.C02.Rsp	Query Next Image Response client command	True	-
OTA.C.C05.Rsp	Image Block Response client command	True	-
OTA.C.C07.Rsp	Upgrade End Response client command	True	-
OTA.C.C09.Rsp	Query Device Specific File Response client command	False	-
OTA.C.C01.Tx	Query Next Image Request client command	True	-
OTA.C.C03.Tx	Image Block Request client command	True	-
OTA.C.C04.Tx	Image Page Request client command	False	-
OTA.C.C06.Tx	Upgrade End Request client command	True	-
OTA.C.C08.Tx	Query Device Specific File Request client command	False	-
OUPC0a	See Test Spec	Depends on customer application	-
OUPC0b	See Test Spec	Depends on customer application	-
OUPC1	See Test Spec	True	-
OUPC2	See Test Spec	True	-
OUPC3	See Test Spec	True	-
OUPC4	See Test Spec	True	-
OUPC5	See Test Spec	False	-
OUPC6	See Test Spec	False	-
OUPC7	See Test Spec	True	-

PICS Code	Name	Supported	Notes
OUPC8	See Test Spec	True	-
OUPC9	See Test Spec	Depends on customer application	-
OUPC10	See Test Spec	Depends on customer application	-
OUPC11	See Test Spec	True	-
OUPC12	See Test Spec	True	-

5.29.2 Over-The-Air upgrade cluster functions

5.29.2.1 *ZbZclOtaClientAlloc*

```
struct ZbZclClusterT * ZbZclOtaClientAlloc(struct ZigBeeT *zb, struct ZbZclOtaClientConfig
*config, void *arg);
```

Create a new instance of the OTA Upgrade Client cluster

Table 434. **ZbZclOtaClientAlloc parameters**

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster
`config`	Pointer to OTA client configuration structure
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: Cluster pointer, or NULL if there is an error.

5.29.2.2 *ZbZclOtaClientDiscover*

```
enum ZclStatusCodeT ZbZclOtaClientDiscover(struct ZbZclClusterT *cluster, const struct
ZbApsAddrT *addr);
```

Discover OTA Upgrade Server

Table 435. **ZbZclOtaClientDiscover parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`addr`	Destination address for discovery

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.29.2.3 *ZbZclOtaClientDiscoverForced*

```
void ZbZclOtaClientDiscoverForced(struct ZbZclClusterT *cluster, uint64_t ieee, uint8_t
endpoint);
```

Set the OTA Upgrade Server directly (without discovery)

Table 436. **ZbZclOtaClientDiscoverForced parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command

Parameter	Description
`ieee`	OTA Upgrade server IEEE address
`endpoint`	Endpoint on which to create cluster

Return: Void

5.29.2.4 *ZbZclOtaClientGetDefaultCallbacks*

```
void ZbZclOtaClientGetDefaultCallbacks(struct ZbZclOtaClientCallbacksT *callbacks);
```

Load the default callbacks for ECDSA Suite 2 support

Table 437. **ZbZclOtaClientGetDefaultCallbacks parameters**

Parameter	Description
`callbacks`	Structure containing any callback function pointers for this cluster

Return: Void

5.29.2.5 *ZbZclOtaClientImageTransferResume*

```
enum ZclStatusCodeT ZbZclOtaClientImageTransferResume(struct ZbZclClusterT*cluster);
```

Resume an OTA Upgrade transfer `ZbZclOtaClientImageTransferResume` is only called by the NHLE if it returned ZCL_STATUS_WAIT_FOR_DATA for the write_`image` callback.

Table 438. **ZbZclOtaClientImageTransferResume parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.29.2.6 *ZbZclOtaClientImageTransferStart*

```
enum ZclStatusCodeT ZbZclOtaClientImageTransferStart(struct ZbZclClusterT*cluster);
```

Initiate a OTA Upgrade transfer

Table 439. **ZbZclOtaClientImageTransferStart parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.29.2.7 *ZbZclOtaClientQueryNextImageReq*

```
enum ZclStatusCodeT ZbZclOtaClientQueryNextImageReq(struct ZbZclClusterT *cluster, struct
ZbZclOtaImageDefinition *image_definition, uint8_t field_control, uint16_t hardware_version);
```

Send a Query Next Image Request command

Table 440. ZbZclOtaClientQueryNextImageReq parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`image_definition`	OTA header image definition structure
`field_control`	Field control
`hardware_version`	Hardware version

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.29.2.8 ZbZclOtaHeaderParse

```
uint8_t ZbZclOtaHeaderParse(const uint8_t *payload, const uint8_t length, struct
ZbZclOtaHeader *header);
```

Parse an OTA Upgrade payload buffer’s header information

Table 441. ZbZclOtaHeaderParse parameters

Parameter	Description
`payload`	OTA upgrade payload buffer
`length`	Length of OTA upgrade payload
`header`	OTA upgrade header fields structure

Return: Number of bytes parsed

5.29.2.9 ZbZclOtaServerAlloc

```
struct ZbZclClusterT * ZbZclOtaServerAlloc(struct ZigBeeT *zb, struct ZbZclOtaServerConfig
*config, void *arg);
```

Create a new instance of the OTA Upgrade Server cluster

Table 442. ZbZclOtaServerAlloc parameters

Parameter	Description
`zb`	Zigbee® stack instance
`config`	OTA upgrade server configuration structure
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: Cluster pointer, or NULL if there is an error.

5.29.2.10 ZbZclOtaServerImageNotifyReq

```
enum ZclStatusCodeT ZbZclOtaServerImageNotifyReq(struct ZbZclClusterT *cluster, const struct
ZbApsAddrT *dst, uint8_t payload_type, uint8_t jitter, struct ZbZclOtaImageDefinition
*image_definition);
```

Send an OTA image notify server command registering an image. This does not automatically send an image notify message, the OTA server application can use `ZbZclOtaServerImageNotifyReq` after registering an image to notify clients of the availability of a new image

Table 443. ZbZclOtaServerImageNotifyReq parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`payload_type`	Payload type
`jitter`	Jitter
`image_definition`	OTA header image definition structure

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.29.2.11 *ZbZclOtaServerUpgradeEndResp*

```
enum ZclStatusCodeT ZbZclOtaServerUpgradeEndResp(struct ZbZclClusterT *cluster, const
struct ZbApsAddrT dst, struct ZbZclOtaImageDefinition *image_definition, struct
ZbZclOtaEndResponseTimes end_response_times);
```

Send an OTA upgrade end response

Table 444. ZbZclOtaServerUpgradeEndResp parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`image_definition`	OTA header image definition structure
`end_response_times`	Upgrade end response command structure

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.29.3 Over-The-Air upgrade cluster enumerations

5.29.3.1 *ZbZclOtaActivationPolicy*

Table 445. OTA Upgrade UpgradeActivationPolicy enumerations

Parameter	Description
ZCL_OTA_ACTIVATION_POLICY_SERVER	OTA server activation allowed
ZCL_OTA_ACTIVATION_POLICY_OUT_OF_BAND	Out-of-band activation only

5.29.3.2 *ZbZclOtaHeaderFieldCtrlBitmask*

Table 446. OTA Upgrade Header Field Control Bitmask enumerations

Parameter	Description
ZCL_OTA_HEADER_FIELD_CONTROL_SECURITY_V ERSION	Security credential version present
ZCL_OTA_HEADER_FIELD_CONTROL_DEVICE_SPE CIFIC	Device specific file
ZCL_OTA_HEADER_FIELD_CONTROL_HARDWARE_ VERSIONS	Hardware versions present

5.29.3.3 ZbZclOtaImageBlkReqFldCtrl

Table 447. OTA Upgrade Image Block Request Field Control Bitmask enumerations

Parameter	Description
ZCL_OTA_IMAGE_BLOCK_FC_IEEE	Request IEEE node address present
ZCL_OTA_IMAGE_BLOCK_FC_MAX_BLOCK	`MinimumBlockPeriod` present

5.29.3.4 ZbZclOtaImageNotifyCmd

Table 448. OTA Upgrade Image Notify Command Payload enumerations

Parameter	Description
ZCL_OTA_NOTIFY_TYPE_JITTER	Query jitter
ZCL_OTA_NOTIFY_TYPE_MFG_CODE	Query jitter and manufacturer code
ZCL_OTA_NOTIFY_TYPE_IMAGE_TYPE	Query jitter, manufacturer code, and image type
ZCL_OTA_NOTIFY_TYPE_FILE_VERSION	Query jitter, manufacturer code, image type, and new file version

5.29.3.5 ZbZclOtaImageType

Table 449. OTA Upgrade Image Types enumerations

Parameter	Description
ZCL_OTA_IMAGE_TYPE_MFG_MIN	Manufacturer specific (start)
ZCL_OTA_IMAGE_TYPE_MFG_MAX	Manufacturer specific (end)
ZCL_OTA_IMAGE_TYPE_CLI_SECURITY_CRED	Client security credentials
ZCL_OTA_IMAGE_TYPE_CLI_CONFIG	Client configuration
ZCL_OTA_IMAGE_TYPE_SERVER_LOG	Server log
ZCL_OTA_IMAGE_TYPE_PICTURE	Picture
ZCL_OTA_IMAGE_TYPE_WILDCARD	Wild card

5.29.3.6 ZbZclOtaQueryFldCtrlHwVer

Table 450. OTA Upgrade Field Control Hardware Version enumerations

Parameter	Description
ZCL_OTA_QUERY_FIELD_CONTROL_HW_VERSION	Hardware version

5.29.3.7 ZbZclOtaSecCredential

Table 451. OTA Upgrade Security Credential Version enumerations

Parameter	Description
ZCL_OTA_SEC_CRED_SE_1_0	SE 1.0
ZCL_OTA_SEC_CRED_SE_1_1	SE 1.1
ZCL_OTA_SEC_CRED_SE_2_0	SE 2.0
ZCL_OTA_SEC_CRED_SE_1_2	SE 1.2

5.29.3.8 *ZbZclOtaStackVersion*

Table 452. **OTA Upgrade Zigbee Stack Version Values**

Parameter	Description
ZCL_OTA_STACK_VERSION_2006	ZigBee 2006
ZCL_OTA_STACK_VERSION_2007	ZigBee 2007
ZCL_OTA_STACK_VERSION_PRO	ZigBee Pro
ZCL_OTA_STACK_VERSION_IP	ZigBee IP

5.29.3.9 *ZbZclOtaStatus*

Table 453. **OTA Upgrade Status Attribute Values**

Parameter	Description
ZCL_OTA_STATUS_NORMAL	Normal
ZCL_OTA_STATUS_DOWNLOAD_IN_PROGRESS	Download in progress
ZCL_OTA_STATUS_DOWNLOAD_COMPLETE	Download complete
ZCL_OTA_STATUS_WAITING_TO_UPGRADE	Waiting to upgrade
ZCL_OTA_STATUS_COUNT_DOWN	Count down
ZCL_OTA_STATUS_WAIT_FOR_MORE	Waiting to upgrade via external event

5.29.3.10 *ZbZclOtaSubElementTag*

Table 454. **OTA Upgrade Tag Identifiers enumerations**

Parameter	Description
ZCL_OTA_SUB_TAG_UPGRADE_IMAGE	Upgrade image
ZCL_OTA_SUB_TAG_ECDSA_SIG1	ECDSA signature (crypto suite 1)
ZCL_OTA_SUB_TAG_ECDSA_CERT_1	ECDSA signing certificate (crypto suite 1)
ZCL_OTA_SUB_TAG_IMAGE_INTEGRITY_CODE	Image integrity code
ZCL_OTA_SUB_TAG_PICTURE_DATA	Picture data
ZCL_OTA_SUB_TAG_ECDSA_SIG2	ECDSA signature (crypto suite 2)
ZCL_OTA_SUB_TAG_ECDSA_CERT_2	ECDSA signing certificate (crypto suite 2)

5.29.3.11 *ZbZclOtaSvrAttrId*

Table 455. **OTA Upgrade server attribute IDs**

Parameter	Description
ZCL_OTA_ATTR_UPGRADE_SERVER_ID	`UpgradeServerID`
ZCL_OTA_ATTR_FILE_OFFSET	`FileOffset` (optional)
ZCL_OTA_ATTR_CURRENT_FILE_VERSION	`CurrentFileVersion` (optional)
ZCL_OTA_ATTR_CURRENT_STACK_VERSION	`CurrentZigBeeStackVersion` (optional)
ZCL_OTA_ATTR_DOWNLOAD_FILE_VERSION	`DownloadedFileVersion` (optional)
ZCL_OTA_ATTR_DOWNLOAD_STACK_VERSION	`DownloadedZigBeeStackVersion` (optional)
ZCL_OTA_ATTR_IMAGE_UPGRADE_STATUS	`ImageUpgradeStatus` (optional)

Parameter	Description
ZCL_OTA_ATTR_MANUFACTURER_ID	`ManufacturerID` (optional)
ZCL_OTA_ATTR_IMAGE_TYPE_ID	`ImageTypeID` (optional)
ZCL_OTA_ATTR_MIN_BLOCK_PERIOD	`MinimumBlockPeriod` (optional)
ZCL_OTA_ATTR_IMAGE_STAMP	`ImageStamp` (optional)
ZCL_OTA_ATTR_UPGRADE_ACTIVATION_POLICY	`UpgradeActivationPolicy` (optional)
ZCL_OTA_ATTR_UPGRADE_TIMEOUT_POLICY	`UpgradeTimeoutPolicy` (optional)

5.29.3.12 *ZbZclOtaTimeoutPolicy*

Table 456. **OTA timeout policy attribute IDs**

Parameter	Description
ZCL_OTA_TIMEOUT_POLICY_APPLY_UPGRADE	Apply Upgrade After Timeout
ZCL_OTA_TIMEOUT_POLICY_DO_NOT_APPLY	Do not Apply Upgrade After Timeout

5.29.4 Over-The-Air upgrade cluster structures

5.29.4.1 *ZbZclOtaClientCallbacksT*

OTA Upgrade callbacks configuration

Table 457. **ZbZclOtaClientCallbacksT parameters**

Parameter	Description
`discover_complete` (callback function pointer)	`enum ZclStatusCodeT (*discover_complete)(struct ZbZclClusterT *cluster, void *arg)` Callback to Server, invoked on discovery of OTA Upgrade Server. If NULL and `ZbZclOtaClientDiscover` is successful, the default handler automatically calls `ZbZclOtaClientQueryNextImageReq`.
`image_notify` (callback function pointer)	`enum ZclStatusCodeT (*image_notify)(struct ZbZclClusterT *cluster,uint8_t payload_type, uint8_t jitter, struct ZbZclOtaImageDefinition *image_definition,struct ZbApsdeDataIndT *data_ind, struct ZbZclHeaderT *zcl_header)` Callback to Server, invoked on receipt of Image Notify Server command If NULL, the OTA Client has default callback handlers to take care of a typical OTA firmware upgrade file.
`query_next` (callback function pointer)	`void (*query_next)(struct ZbZclClusterT *cluster, enum ZclStatusCodeT status,struct ZbZclOtaImageDefinition*image_definition, uint32_t image_size, void *arg)` Callback to Server, invoked on receipt of Query Next Image Response command Void return
`update_raw` (callback function pointer)	`enum ZclStatusCodeT (*update_raw)(struct ZbZclClusterT *cluster, uint8_t length, uint8_t *data, void *arg)` Raw image data is sent through this callback, to update running hash of image for example. ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.
`write_tag` (callback function pointer)	`enum ZclStatusCodeT (*write_tag)(struct ZbZclClusterT *cluster, struct ZbZclOtaHeader *header,uint16_t tag_id, uint32_t tag_length, uint8_t data_length, uint8_t *data, void *arg)` Callback to write tag sub-element information ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

Parameter	Description
`write_image` (callback function pointer)	`enum ZclStatusCodeT (*write_image)(struct ZbZclClusterT *cluster, struct ZbZclOtaHeader *header,uint8_t length, uint8_t *data, void *arg)` If set, called by the 'write_tag' default handler (`ZbZclOtaClientImageWriteTagCb`) for ZCL_OTA_SUB_TAG_UPGRADE_IMAGE data. The return status codes tell the OTA Client what to do next: ZCL_STATUS_SUCCESS : request the next block ZCL_STATUS_WAIT_FOR_DATA : wait before requesting the next block. Anything else: abort the OTA download ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.
`image_validate` (callback function pointer)	`enum ZclStatusCodeT (*image_validate)(struct ZbZclClusterT *cluster, struct ZbZclOtaHeader *header, void *arg)` If NULL, provide the ca_pub_key_array in `ZbZclOtaClientConfig`. ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.
`upgrade_end` (callback function pointer)	`enum ZclStatusCodeT (*upgrade_end)(struct ZbZclClusterT *cluster, struct ZbZclOtaHeader *header,uint32_t current_time, uint32_t upgrade_time, void *arg)` Callback to handle Upgrade End Request command. ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.
`reboot` (callback function pointer)	`void (*reboot)(struct ZbZclClusterT *cluster, void *arg)` Shall not be NULL. If set, called by the 'upgrade_end' default handler if successful. ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.
`abort_download` (callback function pointer)	`enum ZclStatusCodeT (*abort_download)(struct ZbZclClusterT *cluster, enum ZbZclOtaCommandId commandId, void *arg)` Abort Download. Shall not be NULL. ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.29.4.2 *ZbZclOtaClientConfig*

OTA Upgrade Client Configuration structure

Table 458. **ZbZclOtaClientConfig parameters**

Parameter	Description
	Profile ID
`uint8_t endpoint`	Endpoint on which the cluster is created on
`enum ZbZclOtaActivationPolicyactivation_policy`	Activation policy
`enum ZbZclOtaTimeoutPolicytimeout_policy`	Timeout policy – value written to ZCL_OTA_ATTR_UPGRADE_TIMEOUT_POLICY
`uint32_t image_block_delay`	Image block delay - milliseconds
`struct ZbZclOtaHeadercurrent_image`	OTA upgrade header fields structure
`uint16_t hardware_version`	Hardware version - Version (0xNN00) | Revision (0x00NN)
`const uint8_t *ca_pub_key_array`	CA public key array
`unsigned int ca_pub_key_len`	CA public key array length
`struct ZbZclOtaClientCallbacksTcallbacks`	Client callback functions

5.29.4.3 *ZbZclOtaEndResponseTimes*

Upgrade End Response command structure

Table 459. ZbZclOtaEndResponseTimes parameters

Parameter	Description
`uint32_t current_Time`	Current time
`uint32_t upgrade_time`	Upgrade time

5.29.4.4 *ZbZclOtaHeader*

OTA Upgrade Header Fields structure

Table 460. ZbZclOtaHeader parameters

Parameter	Description
`uint32_t file_identifier`	OTA upgrade file identifier
`uint16_t header_version`	OTA header version
`uint16_t header_length`	OTA header length
`uint16_t header_field_control`	OTA header field control
`uint16_t manufacturer_code`	Manufacturer code
`uint16_t image_type`	Image type
`uint32_t file_version`	File version
`uint16_t stack_version`	Zigbee® stack version
`uint8_t header_string`	OTA header string
`uint32_t total_image_size`	Total image size (including header)
`uint8_t sec_credential_version`	Security credential version (optional)
`uint64_t file_destination`	Upgrade file destination (optional)
`uint16_t min_hardware_version`	Minimum hardware version (optional)
`uint16_t max_hardware_version`	Maximum hardware version (optional)

5.29.4.5 *ZbZclOtaImageData*

OTA Upgrade Image Data structure

Table 461. ZbZclOtaImageData parameters

Parameter	Description
`uint32_t file_offset`	File offset
`uint8_t data_size`	Data size
`uint8_t data`	Data

5.29.4.6 *ZbZclOtaImageDefinition*

OTA Header Image Definition structure

Table 462. ZbZclOtaImageDefinition parameters

Parameter	Description
`uint16_t manufacturer_code`	Manufacturer code
`uint16_t image_type`	Image type
`uint32_t file_version`	File version

5.29.4.7 *ZbZclOtaImageWaitForData*

Image Block Response Command Payload with WAIT_FOR_DATA status structure

Table 463. ZbZclOtaImageWaitForData parameters

Parameter	Description
`uint32_t current_Time`	Current time
`uint32_t request_time`	Request time
`uint16_t minimum_block_period`	`MinimumBlock` period

5.29.4.8 *ZbZclOtaServerConfig*

OTA Upgrade Server Configuration structure

Table 464. ZbZclOtaServerConfig parameters

Parameter	Description
	Profile ID
`uint8_t endpoint`	Endpoint on which the cluster is created on
`uint16_t minimum_block_period`	Minimum block period
`uint32_t upgrade_end_current_time`	Upgrade end current time
`uint32_t upgrade_end_upgrade_time`	Upgrade end upgrade time
`ZbZclOtaServerImageEvalT image_eval`	Callback to determine if a suitable image exists
`ZbZclOtaServerImageReadT image_read`	Callback read image data from the server
`ZbZclOtaServerUpgradeEndReqT image_upgrade_end_req`	Callback to handle client Upgrade End request
`void *arg`	Pointer to application data that is provided back to the callback function when invoked

5.30 Poll control cluster

```
#include "zcl/general/zcl.poll.control.h"
```

5.30.1 Poll control cluster description

Table 465. Poll control cluster PICS code description

PICS Code	Name	Supported	Notes
POLL.S	Server	True	-
POLL.C	Client	True	-
POLL.S.A0000	`Check-inInterval` server attribute	True	-
POLL.S.A0001	`LongPollInterval` server attribute	True	-
POLL.S.A0002	`ShortPollInterval` server attribute	True	-
POLL.S.A0003	`FastPollTimeout` server attribute	True	-
POLL.S.A0004	`Check-inIntervalMin` server attribute	True	Optional
POLL.S.A0005	`LongPollIntervalMin` server attribute	True	Optional
POLL.S.A0006	`FastPollTimeoutMax` server attribute	True	Optional
POLL.S.AFFFD	`ClusterRevision` server attribute	True	-

PICS Code	Name	Supported	Notes
POLL.S.AFFFE	`AttributeReportingStatus` server attribute	False	-
POLL.S.C00.Rsp	Check-in Response server command	True	-
POLL.S.C01.Rsp	Fast Poll Stop server command	True	-
POLL.S.C02.Rsp	Set Long Poll Interval server command	True	-
POLL.S.C03.Rsp	Set Short Poll Interval server command	True	-
POLL.S.C00.Tx	Check-in server command	True	-
POLL.C.AFFFD	`ClusterRevision` client attribute	True	-
POLL.C.AFFFE	`AttributeReportingStatus` client attribute	False	-
POLL.C.C00.Rsp	Check-in client command	True	-
POLL.C.C00.Tx	Check-in Response client command	True	-
POLL.C.C01.Tx	Fast Poll Stop client command	True	-
POLL.C.C02.Tx	Set Long Poll Interval client command	True	-
POLL.C.C03.Tx	Set Short Poll Interval client command	True	-

5.30.2 Poll control cluster functions

5.30.2.1 *zcl_poll_client_alloc*

```
struct ZbZclClusterT * zcl_poll_client_alloc(struct ZigBeeT *zb, uint8_t endpoint, struct
ZbZclPollControlClientCallbackT *callbacks, void *arg);
```

Create a new instance of the Poll Control Client cluster

Table 466. **zcl_poll_client_alloc parameters**

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster
`callbacks`	Structure containing any callback function pointers for this cluster
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: Cluster pointer, or NULL if there is an error.

5.30.2.2 *zcl_poll_client_set_checkin_rsp*

```
enum ZclStatusCodeT zcl_poll_client_set_checkin_rsp(struct ZbZclClusterT *cluster, struct
ZbZclPollControlClientCheckinInfo *info);
```

Set Check-in Response configuration

Table 467. **zcl_poll_client_set_checkin_rsp parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`info`	Check-in configuration info

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.30.2.3 *zcl_poll_client_set_long_intvl_req*

```
enum ZclStatusCodeT zcl_poll_client_set_long_intvl_req(struct ZbZclClusterT*cluster, struct
ZbZclPollControlClientSetLongReq *req, void (*callback)(struct ZbZclCommandRspT *rsp, void
*arg), void *arg);
```

Send a Set Long Interval command

Table 468. zcl_poll_client_set_long_intvl_req parameters

Parameter	Description
cluster	Cluster instance from which to send this command
req	Set Long Interval command request structure
callback	Callback function that is invoked when the response is received
arg	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.30.2.4 *zcl_poll_client_set_short_intvl_req*

```
enum ZclStatusCodeT zcl_poll_client_set_short_intvl_req(struct ZbZclClusterT*cluster, struct
ZbZclPollControlClientSetShortReq *req, void (*callback)(struct ZbZclCommandRspT *rsp, void
*arg), void *arg);
```

Send a Set Short Interval command

Table 469. zcl_poll_client_set_short_intvl_req parameters

Parameter	Description
cluster	Cluster instance from which to send this command
req	Set Short Interval command request structure
callback	Callback function that is invoked when the response is received
arg	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.30.2.5 *zcl_poll_client_stop_fastpoll_req*

```
enum ZclStatusCodeT zcl_poll_client_stop_fastpoll_req(struct ZbZclClusterT*cluster, struct
ZbZclPollControlClientStopReq *req, void (*callback)(struct ZbZclCommandRspT *rsp, void
*arg), void *arg);
```

Send a Fast Poll Stop command

Table 470. zcl_poll_client_stop_fastpoll_req parameters

Parameter	Description
cluster	Cluster instance from which to send this command
req	Fast Poll Stop command request structure
callback	Callback function that is invoked when the response is received
arg	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.30.2.6 *zcl_poll_server_alloc*

```
struct ZbZclClusterT * zcl_poll_server_alloc(struct ZigBeeT *zb, uint8_t endpoint, struct
ZbZclPollControlServerCallbackT *callbacks, void *arg);
```

Create a new instance of the Poll Control Server cluster

Table 471. **zcl_poll_server_alloc parameters**

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster
`callbacks`	Structure containing any callback function pointers for this cluster
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: Cluster pointer, or NULL if there is an error.

5.30.2.7 *zcl_poll_server_send_checkin*

```
enum ZclStatusCodeT zcl_poll_server_send_checkin(struct ZbZclClusterT *cluster);
```

Send a Check-in command to any bound Clients.

Table 472. **zcl_poll_server_send_checkin parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.30.2.8 *zcl_poll_server_write_long_poll_intvl*

```
enum ZclStatusCodeT zcl_poll_server_write_long_poll_intvl(struct ZbZclClusterT*clusterPtr,
uint32_t long_poll_intvl);
```

Change the long polling interval. Default is disabled (ZCL_INVALID_UNSIGNED_32BIT = 0xFFFFFFFF).

Table 473. **zcl_poll_server_write_long_poll_intvl parameters**

Parameter	Description
`clusterPtr`	Cluster instance from which to send this command
`long_poll_intvl`	Polling interval in quarter-seconds (1 = 250 mS)

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.30.3 Poll control cluster enumerations

5.30.3.1 *ZbZclPollControlSvrAttrT*

Table 474. **Poll Control server attribute IDs**

Parameter	Description
ZCL_POLL_CHECK_IN_INTERVAL	`Check-inInterval`
ZCL_POLL_LONG_POLL_INTERVAL	`LongPollInterval`

Parameter	Description
ZCL_POLL_SHORT_POLL_INTERVAL	`ShortPollInterval`
ZCL_POLL_FAST_POLL_TIMEOUT	`FastPollTimeout`
ZCL_POLL_CHECK_IN_INTERVAL_MIN	`Check-inIntervalMin` (optional)
ZCL_POLL_LONG_POLL_INTERVAL_MIN	`LongPollIntervalMin` (optional)
ZCL_POLL_FAST_POLL_TIMEOUT_MAX	`FastPollTimeoutMax` (optional)

5.30.4 Poll control cluster structures

5.30.4.1 *ZbZclPollControlClientCallbackT*

Poll Control Client callbacks configuration

Table 475. ZbZclPollControlClientCallbackT parameters

Parameter	Description
`checkin_rsp_callback` (callback function pointer)	`Enum ZclStatusCodeT (*checkin_rsp_callback)(struct ZbZclClusterT *clusterPtr,struct zcl_poll_checkin_rsp_t *rsp_info, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of Check-in Response command

5.30.4.2 *ZbZclPollControlClientCheckinInfo*

Check-in Info structure

Table 476. ZbZclPollControlClientCheckinInfo parameters

Parameter	Description
`struct ZbApsAddrT dst`	Destination Address
`bool start_fast_poll`	Start fast poll
`uint16_t fast_poll_timeout`	Fast poll timeout

5.30.4.3 *ZbZclPollControlClientSetLongReq*

Set Long Poll Interval command structure

Table 477. ZbZclPollControlClientSetLongReq parameters

Parameter	Description
`struct ZbApsAddrT dst`	Destination Address
`uint32_t interval`	New long poll interval

5.30.4.4 *ZbZclPollControlClientSetShortReq*

Set Short Poll Interval command structure

Table 478. ZbZclPollControlClientSetShortReq parameters

Parameter	Description
`struct ZbApsAddrT dst`	Destination Address
`uint16_t interval`	New short poll interval

5.30.4.5 *ZbZclPollControlClientStopReq*

Fast Poll Stop command structure

Table 479. ZbZclPollControlClientStopReq parameters

Parameter	Description
`struct ZbApsAddrT dst`	Destination Address

5.30.4.6 *ZbZclPollControlServerCallbackT*

Poll Control Server callbacks configuration

Table 480. ZbZclPollControlServerCallbackT parameters

Parameter	Description
`checkin_rsp` (callback function pointer)	`Void (*checkin_rsp)(struct ZbZclClusterT *clusterPtr,struct zcl_poll_checkin_rsp_t *rsp_info, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of Check-in Response command

5.30.4.7 *zcl_poll_checkin_rsp_t*

Check-in Response command structure

Table 481. zcl_poll_checkin_rsp_t parameters

Parameter	Description
`enum ZclStatusCodeT status`	Status
`bool start_fast_poll`	Start fast polling
`uint16_t fast_poll_timeout`	Fast poll timeout

5.31 Power configuration cluster

```
#include "zcl/general/zcl.power.config.h"
```

5.31.1 Power configuration cluster description

Table 482. Power configuration cluster PICS code description

PICS Code	Name	Supported	Notes
PC.S	Server	True	-
PC.C	Client	True	-
PC.S.A0000	`MainsVoltage` server attribute	True	Optional
PC.S.A0001	`MainsFrequency` server attribute	True	Optional
PC.S.A0010	`MainsAlarmMask` server attribute	True	Optional
PC.S.A0011	`MainsVoltageMinThreshold` server attribute	True	Optional
PC.S.A0012	`MainsVoltageMaxThreshold` server attribute	True	Optional
PC.S.A0020	`BatteryVoltage` server attribute	True	Optional
PC.S.A0021	`BatteryPercentageRemaining` server attribute	True	Optional
PC.S.A0030	`BatteryManufacturer` server attribute	True	Optional

PICS Code	Name	Supported	Notes
PC.S.A0031	BatterySize server attribute	True	Optional
PC.S.A0032	BatteryAHrRating server attribute	True	Optional
PC.S.A0033	BatteryQuantity server attribute	True	Optional
PC.S.A0034	BatteryRatedVoltage server attribute	True	Optional
PC.S.A0035	BatteryAlarmMask server attribute	True	Optional
PC.S.A0036	BatteryVoltageMinThreshold server attribute	True	Optional
PC.S.A003a	BatteryPercentageMinThreshold server attribute	True	Optional
PC.S.A003e	BatteryAlarmState server attribute	True	Optional
PC.S.A0051	BatterySize - Second battery server attribute	True	Optional
PC.S.A0071	BatterySize - Third Battery server attribute	True	Optional
PC.S.Afffd	ClusterRevision server attribute	True	-
PC.S.Afffe	AttributeReportingStatus server attribute	False	-
PC.C.Afffd	ClusterRevision client attribute	True	-
PC.C.Afffe	AttributeReportingStatus client attribute	False	-

5.31.2 Power configuration cluster functions

5.31.2.1 *ZbZclPowerConfigClientAlloc*

```
struct ZbZclClusterT * ZbZclPowerConfigClientAlloc(struct ZigBeeT *zb, uint8_t endpoint);
```

Create a new instance of the Power Configuration Client cluster

Table 483. ZbZclPowerConfigClientAlloc parameters

Parameter	Description
zb	Zigbee® stack instance
endpoint	Endpoint on which to create cluster

Return: Cluster pointer, or NULL if there is an error.

5.31.2.2 *ZbZclPowerConfigServerAlloc*

```
struct ZbZclClusterT * ZbZclPowerConfigServerAlloc(struct ZigBeeT *zb, uint8_t endpoint);
```

Create a new instance of the Power Configuration Server cluster

Table 484. ZbZclPowerConfigServerAlloc parameters

Parameter	Description
zb	zb
endpoint	Endpoint on which to create cluster

Return: Cluster pointer, or NULL if there is an error.

5.31.3 Power configuration cluster enumerations

5.31.3.1 ZbZclPowerConfigSvrAttrT

Table 485. Power Configuration server attributes IDs

Parameter	Description
ZCL_POWER_CONFIG_ATTR_MAINS_VOLTAGE	MainsVoltage (optional)
ZCL_POWER_CONFIG_ATTR_MAINS_FREQ	MainsFrequency (optional)
ZCL_POWER_CONFIG_ATTR_MAINS_ALARM_MASK	MainsAlarmMask (optional)
ZCL_POWER_CONFIG_ATTR_MAINS_VOLT_MIN	MainsVoltageMinThreshold (optional)
ZCL_POWER_CONFIG_ATTR_MAINS_VOLT_MAX	MainsVoltageMaxThreshold (optional)
ZCL_POWER_CONFIG_ATTR_MAINS_VOLT_DWELL	MainsVoltageDwellTripPoint (optional)
ZCL_POWER_CONFIG_ATTR_BATTERY_VOLTAGE	BatteryVoltage (optional)
ZCL_POWER_CONFIG_ATTR_BATTERY_PCT	BatteryPercentageRemaining (optional)
ZCL_POWER_CONFIG_ATTR_BATTERY_MFR_NAME	BatteryManufacturer (optional)
ZCL_POWER_CONFIG_ATTR_BATTERY_SIZE	BatterySize (optional)
ZCL_POWER_CONFIG_ATTR_BATTERY_AHRRATING	BatteryAHrRating (optional)
ZCL_POWER_CONFIG_ATTR_BATTERY_QUANTITY	BatteryQuantity (optional)
ZCL_POWER_CONFIG_ATTR_BATTERY_RATED_VO LT	BatteryRatedVoltage (optional)
ZCL_POWER_CONFIG_ATTR_BATTERY_ALARM_MA SK	BatteryAlarmMask (optional)
ZCL_POWER_CONFIG_ATTR_BATTERY_VOLT_MIN	BatteryVoltageMinThreshold (optional)
ZCL_POWER_CONFIG_ATTR_BATTERY_VTHRESHO LD1	BatteryVoltageThreshold1 (optional)
ZCL_POWER_CONFIG_ATTR_BATTERY_VTHRESHO LD2	BatteryVoltageThreshold2 (optional)
ZCL_POWER_CONFIG_ATTR_BATTERY_VTHRESHO LD3	BatteryVoltageThreshold3 (optional)
ZCL_POWER_CONFIG_ATTR_BATTERY_PCT_MIN	BatteryPercentageMinThreshold (optional)
ZCL_POWER_CONFIG_ATTR_BATTERY_PTHRESHO LD1	BatteryPercentageThreshold1 (optional)
ZCL_POWER_CONFIG_ATTR_BATTERY_PTHRESHO LD2	BatteryPercentageThreshold2 (optional)
ZCL_POWER_CONFIG_ATTR_BATTERY_PTHRESHO LD3	BatteryPercentageThreshold3 (optional)
ZCL_POWER_CONFIG_ATTR_BATTERY_ALARM_ST ATE	BatteryAlarmState (optional)
ZCL_POWER_CONFIG_ATTR_BATTERY2_VOLTAGE	BatteryVoltage - Second battery (optional)
ZCL_POWER_CONFIG_ATTR_BATTERY2_PCT	BatteryPercentageRemaining - Second battery (optional)
ZCL_POWER_CONFIG_ATTR_BATTERY2_MFR_NAME	BatteryManufacturer - Second battery (optional)
ZCL_POWER_CONFIG_ATTR_BATTERY2_SIZE	BatterySize - Second battery (optional)
ZCL_POWER_CONFIG_ATTR_BATTERY2_AHRRATIN G	BatteryAHrRating - Second battery (optional)
ZCL_POWER_CONFIG_ATTR_BATTERY2_QUANTITY	BatteryQuantity - Second battery (optional)
ZCL_POWER_CONFIG_ATTR_BATTERY2_RATED_V OLT	BatteryRatedVoltage - Second battery (optional)
ZCL_POWER_CONFIG_ATTR_BATTERY2_ALARM_M ASK	BatteryAlarmMask - Second battery (optional)
ZCL_POWER_CONFIG_ATTR_BATTERY2_VOLT_MIN	BatteryVoltageMinThreshold - Second battery (optional)
ZCL_POWER_CONFIG_ATTR_BATTERY2_VTHRESH OLD1	BatteryVoltageThreshold1 - Second battery (optional)
ZCL_POWER_CONFIG_ATTR_BATTERY2_VTHRESH OLD2	BatteryVoltageThreshold2 - Second battery (optional)

Parameter	Description
ZCL_POWER_CONFIG_ATTR_BATTERY2_VTHRESH OLD3	BatteryVoltageThreshold3 - Second battery (optional)
ZCL_POWER_CONFIG_ATTR_BATTERY2_PCT_MIN	BatteryPercentageMinThreshold - Second battery (optional)
ZCL_POWER_CONFIG_ATTR_BATTERY2_PTHRESH OLD1	BatteryPercentageThreshold1 - Second battery (optional)
ZCL_POWER_CONFIG_ATTR_BATTERY2_PTHRESH OLD2	BatteryPercentageThreshold2 - Second battery (optional)
ZCL_POWER_CONFIG_ATTR_BATTERY2_PTHRESH OLD3	BatteryPercentageThreshold3 - Second battery (optional)
ZCL_POWER_CONFIG_ATTR_BATTERY2_ALARM_ST ATE	BatteryAlarmState - Second battery (optional)
ZCL_POWER_CONFIG_ATTR_BATTERY3_VOLTAGE	BatteryVoltage - Third Battery (optional)
ZCL_POWER_CONFIG_ATTR_BATTERY3_PCT	BatteryPercentageRemaining - Third Battery (optional)
ZCL_POWER_CONFIG_ATTR_BATTERY3_MFR_NAME	BatteryManufacturer - Third Battery (optional)
ZCL_POWER_CONFIG_ATTR_BATTERY3_SIZE	BatterySize - Third Battery (optional)
ZCL_POWER_CONFIG_ATTR_BATTERY3_AHRRATIN G	BatteryAHrRating - Third Battery (optional)
ZCL_POWER_CONFIG_ATTR_BATTERY3_QUANTITY	BatteryQuantity - Third Battery (optional)
ZCL_POWER_CONFIG_ATTR_BATTERY3_RATED_V OLT	BatteryRatedVoltage - Third Battery (optional)
ZCL_POWER_CONFIG_ATTR_BATTERY3_ALARM_M ASK	BatteryAlarmMask - Third Battery (optional)
ZCL_POWER_CONFIG_ATTR_BATTERY3_VOLT_MIN	BatteryVoltageMinThreshold - Third Battery (optional)
ZCL_POWER_CONFIG_ATTR_BATTERY3_VTHRESH OLD1	BatteryVoltageThreshold1 - Third Battery (optional)
ZCL_POWER_CONFIG_ATTR_BATTERY3_VTHRESH OLD2	BatteryVoltageThreshold2 - Third Battery (optional)
ZCL_POWER_CONFIG_ATTR_BATTERY3_VTHRESH OLD3	BatteryVoltageThreshold3 - Third Battery (optional)
ZCL_POWER_CONFIG_ATTR_BATTERY3_PCT_MIN	BatteryPercentageMinThreshold - Third Battery (optional)
ZCL_POWER_CONFIG_ATTR_BATTERY3_PTHRESH OLD1	BatteryPercentageThreshold1 - Third Battery (optional)
ZCL_POWER_CONFIG_ATTR_BATTERY3_PTHRESH OLD2	BatteryPercentageThreshold2 - Third Battery (optional)
ZCL_POWER_CONFIG_ATTR_BATTERY3_PTHRESH OLD3	BatteryPercentageThreshold3 - Third Battery (optional)
ZCL_POWER_CONFIG_ATTR_BATTERY3_ALARM_ST ATE	BatteryAlarmState - Third Battery (optional)

5.32 Power profile cluster

```
#include "zcl/general/zcl.power.profile.h"
```

5.32.1 Power profile cluster description

Table 486. Power profile cluster PICS code definition

PICS Code	Name	Supported	Notes
PWR.S	Server	True	-
PWR.C	Client	True	-

PICS Code	Name	Supported	Notes
PWR.S.A0000	TotalProfileNum server attribute	True	-
PWR.S.A0001	MultipleScheduling server attribute	True	-
PWR.S.A0002	EnergyFormatting server attribute	True	-
PWR.S.A0003	EnergyRemote server attribute	True	-
PWR.S.A0004	ScheduleMode server attribute	True	-
PWR.S.A0004.Report.Tx	ScheduleMode server attribute reports	True	-
PWR.S.Afffd	ClusterRevision server attribute	True	-
PWR.S.Affe	AttributeReportingStatus server attribute	False	-
PWR.S.C00.Rsp	PowerProfileRequest server command	True	-
PWR.S.C01.Rsp	PowerProfileStateRequest server command	True	-
PWR.S.C02.Rsp	GetPowerProfilePriceResponse server command	True	-
PWR.S.C03.Rsp	GetOverallSchedulePriceResponse server command	True	-
PWR.S.C04.Rsp	EnergyPhasesScheduleNotification server command	True	-
PWR.S.C05.Rsp	EnergyPhasesScheduleResponse server command	True	-
PWR.S.C06.Rsp	PowerProfileScheduleConstraintsRequest server command	True	-
PWR.S.C07.Rsp	EnergyPhasesScheduleStateRequest server command	True	-
PWR.S.C08.Rsp	GetPowerProfilePriceExtendedResponse server command	True	-
PWR.S.C00.Tx	PowerProfileNotification server command	True	-
PWR.S.C01.Tx	PowerProfileResponse server command	True	-
PWR.S.C02.Tx	PowerProfileStateResponse server command	True	-
PWR.S.C03.Tx	GetPowerProfilePrice server command	True	-
PWR.S.C04.Tx	PowerProfilesStateNotification server command	True	-
PWR.S.C05.Tx	GetOverallSchedulePrice server command	True	-
PWR.S.C06.Tx	EnergyPhasesScheduleRequest server command	True	-
PWR.S.C07.Tx	EnergyPhasesScheduleStateResponse server command	True	-
PWR.S.C08.Tx	EnergyPhasesScheduleStateNotification server command	True	-
PWR.S.C09.Tx	EnergyPhasesScheduleStateNotification server command	True	-
PWR.S.C0a.Tx	PowerProfileScheduleConstraintsResponse server command	True	-
PWR.S.C0b.Tx	GetPowerProfilePriceExtended server command	True	-
PWR.C.A0004.Report.Rsp	ScheduleMode client attribute reports	False	-
PWR.C.Afffd	ClusterRevision client attribute	True	-
PWR.C.Affe	AttributeReportingStatus client attribute	False	-
PWR.C.C00.Rsp	PowerProfileNotification client command	True	-
PWR.C.C01.Rsp	PowerProfileResponse client command	True	-
PWR.C.C02.Rsp	PowerProfileStateResponse client command	True	-
PWR.C.C03.Rsp	GetPowerProfilePrice client command	True	-
PWR.C.C04.Rsp	PowerProfilesStateNotification client command	True	-
PWR.C.C05.Rsp	GetOverallSchedulePrice client command	True	-

PICS Code	Name	Supported	Notes
PWR.C.C06.Rsp	`EnergyPhasesScheduleRequest` client command	True	-
PWR.C.C07.Rsp	`EnergyPhasesScheduleStateResponse` client command	True	-
PWR.C.C08.Rsp	`EnergyPhasesScheduleStateNotification` client command	True	-
PWR.C.C09.Rsp	`EnergyPhasesScheduleStateNotification` client command	True	-
PWR.C.C0a.Rsp	`PowerProfileScheduleConstraintsResponse` client command	True	-
PWR.C.C0b.Rsp	`GetPowerProfilePriceExtended` client command	True	-
PWR.C.C00.Tx	`PowerProfileRequest` client command	True	-
PWR.C.C01.Tx	`PowerProfileStateRequest` client command	True	-
PWR.C.C02.Tx	`GetPowerProfilePriceResponse` client command	True	-
PWR.C.C03.Tx	`GetOverallSchedulePriceResponse` client command	True	-
PWR.C.C04.Tx	`EnergyPhasesScheduleNotification` client command	True	-
PWR.C.C05.Tx	`EnergyPhasesScheduleResponse` client command	True	-
PWR.C.C06.Tx	`PowerProfileScheduleConstraintsRequest` client command	True	-
PWR.C.C07.Tx	`EnergyPhasesScheduleStateRequest` client command	True	-
PWR.C.C08.Tx	`GetPowerProfilePriceExtendedResponse` client command	True	-

5.32.2 Power profile cluster functions

5.32.2.1 *ZbZclPowerProfClientAlloc*

```
struct ZbZclClusterT * ZbZclPowerProfClientAlloc(struct ZigBeeT *zb, uint8_t endpoint,
struct ZbZclPowerProfClientCallbacks *callbacks, void *arg);
```

Create a new instance of the Power Profile Client cluster

Table 487. **ZbZclPowerProfClientAlloc parameters**

Parameter	Description
`zb`	`zb`
`endpoint`	Endpoint on which to create cluster
`callbacks`	Structure containing any callback function pointers for this cluster
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: Cluster pointer, or NULL if there is an error.

5.32.2.2 *ZbZclPowerProfClientPhasesNotify*

```
enum ZclStatusCodeT ZbZclPowerProfClientPhasesNotify(struct ZbZclClusterT *cluster, const
struct ZbApsAddrT *dst, struct ZbZclPowerProfCliPhasesNotify *notify, void (*callback)
(struct ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send an Energy Phases Schedule Notification command

Table 488. **ZbZclPowerProfClientPhasesNotify parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address

Parameter	Description
`notify`	Energy phases schedule n`callback`otification command structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.32.2.3 *ZbZclPowerProfClientPhasesResponse*

```
enum ZclStatusCodeT ZbZclPowerProfClientPhasesResponse(struct ZbZclClusterT*cluster, struct
ZbZclAddrInfoT *dst, struct ZbZclPowerProfCliPhasesNotify *notify, void (*callback)(struct
ZbApsdeDataConfT *conf, void *arg), void *arg);
```

Send a `EnergyPhasesScheduleResponse` command

Table 489. ZbZclPowerProfClientPhasesResponse parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`rsp`	`EnergyPhasesScheduleResponse` command response structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.32.2.4 *ZbZclPowerProfClientPhasesSchedStateReq*

```
enum ZclStatusCodeT ZbZclPowerProfClientPhasesSchedStateReq(struct ZbZclClusterT*cluster,
const struct ZbApsAddrT *dst, struct ZbZclPowerProfCliProfileReq *req, void (*callback)
(struct ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a `EnergyPhasesScheduleStateRequest` command

Table 490. ZbZclPowerProfClientPhasesSchedStateReq parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`rsp`	`EnergyPhasesScheduleStateRequest` command request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.32.2.5 *ZbZclPowerProfClientPriceExtRsp*

```
enum ZclStatusCodeT ZbZclPowerProfClientPriceExtRsp(struct ZbZclClusterT *cluster, struct
ZbZclAddrInfoT *dst, struct ZbZclPowerProfCliPriceRsp *rsp, void (*callback)(struct
ZbApsdeDataConfT *conf, void *arg), void *arg);
```

Send a `GetPowerProfilePriceExtendedResponse` command

Table 491. ZbZclPowerProfClientPriceExtRsp parameters

Parameter	Description
cluster	Cluster instance from which to send this command
dst	Requested destination address
rsp	GetPowerProfilePriceExtendedResponse command response structure
callback	Callback function that is invoked when the response is received
arg	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other ZclStatusCodeT value on error.

5.32.2.6 *ZbZclPowerProfClientPriceRsp*

```
enum ZclStatusCodeT ZbZclPowerProfClientPriceRsp(struct ZbZclClusterT *cluster, struct
ZbZclAddrInfoT *dst, struct ZbZclPowerProfCliPriceRsp *rsp, void (*callback)(struct
ZbApsdeDataConfT *conf, void *arg), void *arg);
```

Send a GetPowerProfilePriceResponse command

Table 492. ZbZclPowerProfClientPriceRsp parameters

Parameter	Description
cluster	Cluster instance from which to send this command
dst	Requested destination address
rsp	GetPowerProfilePriceResponse command response structure
callback	Callback function that is invoked when the response is received
arg	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other ZclStatusCodeT value on error.

5.32.2.7 *ZbZclPowerProfClientProfileReq*

```
enum ZclStatusCodeT ZbZclPowerProfClientProfileReq(struct ZbZclClusterT *cluster, const
struct ZbApsAddrT *dst, struct ZbZclPowerProfCliProfileReq *req, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a PowerProfileRequest command

Table 493. ZbZclPowerProfClientProfileReq parameters

Parameter	Description
cluster	Cluster instance from which to send this command
dst	Requested destination address
req	PowerProfileRequest command request structure
callbacks	Structure containing any callback function pointers for this cluster
arg	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other ZclStatusCodeT value on error.

5.32.2.8 ZbZclPowerProfClientSchedConsReq

```
enum ZclStatusCodeT ZbZclPowerProfClientSchedConsReq(struct ZbZclClusterT *cluster, const
struct ZbApsAddrT *dst, struct ZbZclPowerProfCliProfileReq *req, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a `PowerProfileScheduleConstraintsRequest` command

Table 494. ZbZclPowerProfClientSchedConsReq parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`req`	`PowerProfileScheduleConstraintsRequest` command request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.32.2.9 ZbZclPowerProfClientSchedPriceRsp

```
enum ZclStatusCodeT ZbZclPowerProfClientSchedPriceRsp(struct ZbZclClusterT*cluster, struct
ZbZclAddrInfoT *dst, struct ZbZclPowerProfCliSchedPriceRsp *rsp, void (*callback)(struct
ZbApsdeDataConfT *conf, void *arg), void *arg);
```

Send a `GetOverallSchedulePriceResponse` command

Table 495. ZbZclPowerProfClientSchedPriceRsp `parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Destination address for response, including sequence number and tx options
`rsp`	`GetOverallSchedulePriceResponse` command response structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.32.2.10 ZbZclPowerProfClientStateReq

```
enum ZclStatusCodeT ZbZclPowerProfClientStateReq(struct ZbZclClusterT *cluster, const struct
ZbApsAddrT *dst, void (*callback)(struct ZbZclCommandRspT *rsp, void*arg), void *arg);
```

Send a PowerProfileStateRequest command

Table 496. ZbZclPowerProfClientStateReq parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Destination address for request
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.32.2.11 *ZbZclPowerProfServerAlloc*

```
struct ZbZclClusterT * ZbZclPowerProfServerAlloc(struct ZigBeeT *zb, uint8_t endpoint,
struct ZbZclPowerProfServerCallbacks *callbacks, void *arg);
```

Create a new instance of the Power Profile Server cluster

Table 497. **ZbZclPowerProfServerAlloc parameters**

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: Cluster pointer, or NULL if there is an error.

5.32.2.12 *ZbZclPowerProfServerConstraintsNotify*

```
enum ZclStatusCodeT ZbZclPowerProfServerConstraintsNotify(struct ZbZclClusterT*cluster,
const struct ZbApsAddrT *dst, struct ZbZclPowerProfSvrConstraintsNotify*notify, void
(*callback)(struct ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a `PowerProfileScheduleConstraintsNotification` command

Table 498. **ZbZclPowerProfServerConstraintsNotify parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Destination address for request
`notify`	`PowerProfileScheduleConstraintsNotification` command structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.32.2.13 *ZbZclPowerProfServerConstraintsRsp*

```
enum ZclStatusCodeT ZbZclPowerProfServerConstraintsRsp(struct ZbZclClusterT*cluster, struct
ZbZclAddrInfoT *dst, struct ZbZclPowerProfSvrConstraintsNotify*rsp, void (*callback)(struct
ZbApsdeDataConfT *conf, void *arg), void *arg);
```

Send a `PowerProfileScheduleConstraintsResponse` command

Table 499. **ZbZclPowerProfServerConstraintsRsp parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Destination address for response, including sequence number and tx options
`rsp`	`PowerProfileScheduleConstraintsResponse` command response structure
`callback`	Callback function that is invoked when the response is received

Parameter	Description
arg	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other ZclStatusCodeT value on error.

5.32.2.14 *ZbZclPowerProfServerGetPriceReq*

```
enum ZclStatusCodeT ZbZclPowerProfServerGetPriceReq(struct ZbZclClusterT *cluster, const
struct ZbApsAddrT *dst, struct ZbZclPowerProfCliProfileReq *req, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a GetPowerProfilePrice command

Table 500. ZbZclPowerProfServerGetPriceReq parameters

Parameter	Description
cluster	Cluster instance from which to send this command
dst	Destination address for request
req	GetPowerProfilePrice command request structure
callback	Callback function that is invoked when the response is received
arg	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other ZclStatusCodeT value on error.

5.32.2.15 *ZbZclPowerProfServerGetPriceReqExtReq]*

```
enum ZclStatusCodeT ZbZclPowerProfServerGetPriceReqExtReq(struct ZbZclClusterT*cluster,
const struct ZbApsAddrT *dst, struct ZbZclPowerProfSvrGetPriceExtReq*req, void (*callback)
(struct ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a GetPowerProfilePriceExtended command

Table 501. ZbZclPowerProfServerGetPriceReqExtReq parameters

Parameter	Description
cluster	Cluster instance from which to send this command
dst	Requested destination address
req	GetPowerProfilePriceExtended command request structure
callback	Callback function that is invoked when the response is received
arg	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other ZclStatusCodeT value on error.

5.32.2.16 *ZbZclPowerProfServerGetSchedPriceReq*

```
enum ZclStatusCodeT ZbZclPowerProfServerGetSchedPriceReq(struct ZbZclClusterT*cluster, const
struct ZbApsAddrT *dst, void (*callback)(struct ZbZclCommandRspT*rsp, void *arg), void *arg);
```

Send a GetOverallSchedulePrice command

Table 502. ZbZclPowerProfServerGetSchedPriceReq parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Destination address for request
`req`	`GetOverallSchedulePrice` command request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.32.2.17 *ZbZclPowerProfServerPhasesNotify*

```
enum ZclStatusCodeT ZbZclPowerProfServerPhasesNotify(struct ZbZclClusterT *cluster, const
struct ZbApsAddrT *dst, struct ZbZclPowerProfSvrPhasesRsp *notify, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a `EnergyPhasesScheduleStateNotification` Command

Table 503. ZbZclPowerProfServerPhasesNotify parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Destination address for request
`notify`	`EnergyPhasesScheduleStateNotification` command structure
`callback`	Callback function that is invoked when the response is received
`arg`	`arg`

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.32.2.18 *ZbZclPowerProfServerPhasesReq*

```
enum ZclStatusCodeT ZbZclPowerProfServerPhasesReq(struct ZbZclClusterT *cluster, const
struct ZbApsAddrT *dst, struct ZbZclPowerProfCliProfileReq *req, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a `EnergyPhasesScheduleRequest` command

Table 504. ZbZclPowerProfServerPhasesReq parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Destination address for request
`req`	`EnergyPhasesScheduleRequest` command request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.32.2.19 *ZbZclPowerProfServerPhasesRsp*

```
enum ZclStatusCodeT ZbZclPowerProfServerPhasesRsp(struct ZbZclClusterT *cluster, struct
ZbZclAddrInfoT *dst, struct ZbZclPowerProfSvrPhasesRsp *rsp, void (*callback)(struct
ZbApsdeDataConfT *conf, void *arg), void *arg);
```

Send a `EnergyPhasesScheduleStateResponse` command

Table 505. **ZbZclPowerProfServerPhasesRsp parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Destination address for response, including sequence number and tx options
`rsp`	`EnergyPhasesScheduleStateResponse` command response structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.32.2.20 *ZbZclPowerProfServerProfileNotify*

```
enum ZclStatusCodeT ZbZclPowerProfServerProfileNotify(struct ZbZclClusterT*cluster, const
struct ZbApsAddrT *dst, struct ZbZclPowerProfSvrProfileRsp *notify, void (*callback)(struct
ZbZclCommandRspT *zcl_rsp, void *arg), void *arg);
```

Send a `PowerProfileNotification` command. It is sent as a ZCL request with a unique sequence number, and can receive a default response if applicable

Table 506. **ZbZclPowerProfServerProfileNotify parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`notify`	`PowerProfileNotification` command structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.32.2.21 *ZbZclPowerProfServerProfileRsp*

```
enum ZclStatusCodeT ZbZclPowerProfServerProfileRsp(struct ZbZclClusterT *cluster, struct
ZbZclAddrInfoT *dst, struct ZbZclPowerProfSvrProfileRsp *rsp, void (*callback)(struct
ZbApsdeDataConfT *conf, void *arg), void *arg);
```

Send a `PowerProfileResponse` command

Table 507. **ZbZclPowerProfServerProfileRsp parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Destination address for response, including sequence number and tx options
`rsp`	`PowerProfileResponse` command response structure
`callback`	Callback function that is invoked when the response is received

Parameter	Description
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.32.2.22 *ZbZclPowerProfServerStateNotify*

```
enum ZclStatusCodeT ZbZclPowerProfServerStateNotify(struct ZbZclClusterT *cluster, const
struct ZbApsAddrT *dst, struct ZbZclPowerProfSvrStateRsp *notify, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a `PowerProfileStateNotification` Command

Table 508. **ZbZclPowerProfServerStateNotify parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Destination address for request
`notify`	`PowerProfileStateNotification` command structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.32.2.23 *ZbZclPowerProfServerStateRsp*

```
enum ZclStatusCodeT ZbZclPowerProfServerStateRsp(struct ZbZclClusterT *cluster, struct
ZbZclAddrInfoT *dst, struct ZbZclPowerProfSvrStateRsp *rsp, void (*callback)(struct
ZbApsdeDataConfT *conf, void *arg), void *arg);
```

Send a `PowerProfileStateResponse` command

Table 509. **ZbZclPowerProfServerStateRsp parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Destination address for response, including sequence number and tx options
`rsp`	`PowerProfileStateResponse` command response structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.32.3 Power profile cluster enumerations

5.32.3.1 *ZbZclPowerProfileSvrAttrT*

Table 510. **Power Profile server attribute IDs**

Parameter	Description
ZCL_POWER_PROF_SVR_ATTR_TOTAL_PROFILENU M	`TotalProfileNum`
ZCL_POWER_PROF_SVR_ATTR_MULTIPLE_SCHED	`MultipleScheduling`

Parameter	Description
ZCL_POWER_PROF_SVR_ATTR_ENERGY_FORMAT	`EnergyFormatting`
ZCL_POWER_PROF_SVR_ATTR_ENERGY_REMOTE	`EnergyRemote`
ZCL_POWER_PROF_SVR_ATTR_SCHEDULE_MODE	`ScheduleMode`

5.32.4 Power profile cluster structures

5.32.4.1 *ZbZclPowerProfCliPhasesNotify*

Energy Phases Schedule Notification command structure

Table 511. ZbZclPowerProfCliPhasesNotify parameters

Parameter	Description
`uint8_t profile_id`	Power profile ID
`uint8_t num_phases`	Number of scheduled phases
`struct ZbZclPowerProfSchedPhasesched_list`	Schedule List

5.32.4.2 *ZbZclPowerProfCliPriceRsp*

`GetPowerProfilePriceResponse` command structure

Table 512. ZbZclPowerProfCliPriceRsp parameters

Parameter	Description
`uint8_t profile_id`	Power profile ID
`uint16_t currency`	Currency
`uint32_t price`	Price
`uint8_t trailing_digit_and_price_tier`	Price trailing digit and price tier

5.32.4.3 *ZbZclPowerProfCliProfileReq*

`PowerProfileRequest` command structure

Table 513. ZbZclPowerProfCliProfileReq parameters

Parameter	Description
`uint8_t profile_id`	Power profile ID

5.32.4.4 *ZbZclPowerProfCliSchedPriceRsp*

`GetOverallSchedulePriceResponse` command structure

Table 514. ZbZclPowerProfCliSchedPriceRsp parameters

Parameter	Description
`uint16_t currency`	Currency
`uint32_t price`	Price
`uint8_t trailing_digit_and_price_tier`	Price trailing digit and price tier

5.32.4.5 *ZbZclPowerProfClientCallbacks*

Power Profile Client callbacks configuration

Table 515. ZbZclPowerProfClientCallbacks parameters

Parameter	Description
profile_notify (callback function pointer)	Enum ZclStatusCodeT (*profile_notify)(struct bZclClusterT *cluster,struct ZbZclPowerProfSvrProfileRsp *notify, struct ZbZclAddrInfoT *srcInfo, void *arg) Callback to application, invoked on receipt of PowerProfileNotification command
get_price (callback function pointer)	Enum ZclStatusCodeT (*get_price)(struct ZbZclClusterT *cluster,struct ZbZclPowerProfCliProfileReq *req, struct ZbZclAddrInfoT *srcInfo, void *arg) Callback to application, invoked on receipt of GetPowerProfilePrice command
state_notify (callback function pointer)	enum ZclStatusCodeT (*state_notify)(struct ZbZclClusterT *cluster,struct ZbZclPowerProfSvrStateRsp *notify, struct ZbZclAddrInfoT *srcInfo, void *arg) Callback to application, invoked on receipt of PowerProfileStateNotification command
get_sched_price (callback function pointer)	enum ZclStatusCodeT (*get_sched_price)(struct ZbZclClusterT *cluster,struct ZbZclAddrInfoT *srcInfo, void *arg) Callback to application, invoked on receipt of GetOverallSchedulePrice command
phases_req (callback function pointer)	Enum ZclStatusCodeT (*phases_req)(struct ZbZclClusterT *cluster,struct ZbZclPowerProfCliProfileReq *req, struct ZbZclAddrInfoT *srcInfo, void *arg) Callback to application, invoked on receipt of EnergyPhasesScheduleRequest command
phases_notify (callback function pointer)	Enum ZclStatusCodeT (*phases_notify)(struct ZbZclClusterT *cluster,struct ZbZclPowerProfSvrPhasesRsp *notify, struct ZbZclAddrInfoT *srcInfo, void *arg) Callback to application, invoked on receipt of EnergyPhasesScheduleStateNotification command
constraints_notify (callback function pointer)	Enum ZclStatusCodeT (*constraints_notify)(struct ZbZclClusterT *cluster,struct ZbZclPowerProfSvrConstraintsNotify *notify, struct ZbZclAddrInfoT *srcInfo, void *arg) Callback to application, invoked on receipt of PowerProfileScheduleConstraintsNotification command
get_price_ext (callback function pointer)	Enum ZclStatusCodeT (*get_price_ext)(struct ZbZclClusterT *cluster,struct ZbZclPowerProfSvrGetPriceExtReq *req, struct ZbZclAddrInfoT *srcInfo, void *arg) Callback to application, invoked on receipt of GetPowerProfilePriceExtended command

5.32.4.6 *ZbZclPowerProfPhase*

Phase structure

Table 516. ZbZclPowerProfPhase parameters

Parameter	Description
uint8_t energy_phase_id	Energy phase ID
uint8_t macro_phase_id	Macro phase ID
uint16_t expect_duration	Expected duration
uint16_t peak_power	Peak power

Parameter	Description
`uint16_t energy`	Energy
`uint16_t max_activation_delay`	`MaxActivationDelay`

5.32.4.7 *ZbZclPowerProfSchedPhase*

Schedule Phase structure

Table 517. ZbZclPowerProfSchedPhase parameters

Parameter	Description
`MaxActivationDelay`	Energy Phase ID
`uint16_t sched_time`	Scheduled Time

5.32.4.8 *ZbZclPowerProfServerCallbacks*

Power Profile Server callbacks configuration

Table 518. ZbZclPowerProfServerCallbacks parameters

Parameter	Description
`profile_req` (callback function pointer)	`enum ZclStatusCodeT (*profile_req)(struct ZbZclClusterT *cluster,struct ZbZclPowerProfCliProfileReq *req, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of `PowerProfileRequest` command
`state_req` (callback function pointer)	`Enum ZclStatusCodeT (*state_req)(struct ZbZclClusterT *cluster,struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of `PowerProfileStateRequest` command
`phases_notify` (callback function pointer)	`Enum ZclStatusCodeT (*phases_notify)(struct ZbZclClusterT *cluster,struct ZbZclPowerProfCliPhasesNotify *req, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of Energy Phases Schedule Notification command
`sched_contraints_req` (callback function pointer)	`enumZclStatusCodeT (*sched_contraints_req)(struct ZbZclClusterT *cluster,struct ZbZclPowerProfCliProfileReq *req, struct ZbZclAddrInfoT *srcInfo, void *arg)` Callback to application, invoked on receipt of `PowerProfileScheduleConstraintsRequest` command
`phases_sched_state_req` (callback function pointer)	`Enum ZclStatusCodeT (*phases_sched_state_req)(struct ZbZclClusterT *cluster,struct ZbZclPowerProfCliProfileReq *req, struct ZbZclAddrInfoT*srcInfo, void *arg)` Callback to application, invoked on receipt of `EnergyPhasesScheduleStateRequest` command

5.32.4.9 *ZbZclPowerProfSvrConstraintsNotify*

`PowerProfileScheduleConstraintsNotification` command structure

Table 519. ZbZclPowerProfSvrConstraintsNotify parameters

Parameter	Description
`uint8_t profile_id`	Power profile ID
`uint16_t start_after`	Start after
`uint16_t stop_before`	Stop before

5.32.4.10 *ZbZclPowerProfSvrGetPriceExtReq*

`GetPowerProfilePriceExtended` command structure

Table 520. ZbZclPowerProfSvrGetPriceExtReq parameters

Parameter	Description
`uint8_t options`	Options, for example ZCL_PWR_PROF_PRICE_EXT_OPT_START_TIME_PRESENT
`uint8_t profile_id`	Power profile ID
`uint16_t start_time`	`PowerProfileStartTime` - optional (ZCL_PWR_PROF_PRICE_EXT_OPT_START_TIME_PRESENT)

5.32.4.11 *ZbZclPowerProfSvrPhasesRsp*

Energy Phases Schedule Notification and EnergyPhasesScheduleResponse command structure

Table 521. ZbZclPowerProfSvrPhasesRsp parameters

Parameter	Description
`uint8_t profile_id`	Power profile ID
`uint8_t num_sched_energy_phases`	Num of scheduled phases

5.32.4.12 *ZbZclPowerProfSvrProfileRsp*

`PowerProfileNotification` and `PowerProfileResponse` command structure

Table 522. ZbZclPowerProfSvrProfileRsp parameters

Parameter	Description
`uint8_t total_profile_num`	Total profile number
`uint8_t profile_id`	Power profile ID
`uint8_t num_transferred_phases`	Number of transferred phases
`struct ZbZclPowerProfPhasephase_list`	Phase list

5.32.4.13 *ZbZclPowerProfSvrStateRsp*

`PowerProfileStateResponse` command structure

Table 523. ZbZclPowerProfSvrStateRsp parameters

Parameter	Description
`uint8_t profile_count`	Power Profile Count
`struct ZbZclPowerProfileRecordrecord_list`	Record List

5.32.4.14 *ZbZclPowerProfileRecord*

Record structure

Table 524. ZbZclPowerProfileRecord parameters

Parameter	Description
`uint8_t profile_id`	Power profile ID
`uint8_t energy_phase_id`	Energy phase ID
`uint8_t remote_control`	`PowerProfileRemoteControl`
`enum ZbZclPowerProfState state`	`PowerProfile` state

5.33 Pressure measurement cluster

```
#include "zcl/general/zcl.press.meas.h"
```

5.33.1 Pressure measurement cluster description

Table 525. :Pressure measurement cluster PICS code description

PICS Code	Name	Supported	Notes
PRS.S	Server	True	-
PRS.C	Client	True	-
PRS.S.A0000	`MeasuredValue` server attribute	True	-
PRS.S.A0000.Report.Tx	`MeasuredValue` server attribute reports	True	-
PRS.S.A0001	`MinMeasuredValue` server attribute	True	-
PRS.S.A0002	`MaxMeasuredValue` server attribute	True	-
PRS.S.Afffd	`ClusterRevision` server attribute	True	-
PRS.S.Afffe	`AttributeReportingStatus` server attribute	False	-
PRS.C.A0000.Report.Rsp	`MeasuredValue` client attribute reports	False	-
PRS.C.Afffd	`ClusterRevision` client attribute	True	-
PRS.C.Afffe	`AttributeReportingStatus` client attribute	False	-

5.33.2 Pressure measurement cluster functions

5.33.2.1 *ZbZclPressMeasClientAlloc*

```
struct ZbZclClusterT * ZbZclPressMeasClientAlloc(struct ZigBeeT *zb, uint8_t endpoint);
```

Create a new instance of the Pressure Measurement Client cluster

Table 526. ZbZclPressMeasClientAlloc parameters

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster

Return: Cluster pointer, or NULL if there is an error.

5.33.2.2 *ZbZclPressMeasServerAlloc*

```
struct ZbZclClusterT * ZbZclPressMeasServerAlloc(struct ZigBeeT *zb, uint8_t endpoint,
int16_t min, int16_t max);
```

Create a new instance of the Pressure Measurement Server cluster

Table 527. ZbZclPressMeasServerAlloc parameters

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster
`min`	Default minimum measured value written to ZCL_PRESS_MEAS_ATTR_MIN_MEAS_VAL attribute during allocation
`max`	Default maximum measured value written to ZCL_PRESS_MEAS_ATTR_MAX_MEAS_VAL attribute during allocation

Return: Cluster pointer, or NULL if there is an error.

5.33.3 Pressure measurement cluster enumerations

5.33.3.1 ZbZclPressMeasSvrAttrT

Table 528. Pressure Measurement Attribute IDs

Parameter	Description
ZCL_PRESS_MEAS_ATTR_MEAS_VAL	Measured value
ZCL_PRESS_MEAS_ATTR_MIN_MEAS_VAL	Minimum measured value
ZCL_PRESS_MEAS_ATTR_MAX_MEAS_VAL	Maximum measured value
ZCL_PRESS_MEAS_ATTR_TOLERANCE	Tolerance (optional)
ZCL_PRESS_MEAS_ATTR_SCALED_VAL	Scaled value (optional)
ZCL_PRESS_MEAS_ATTR_MIN_SCALED_VAL	Minimum scaled value (optional)
ZCL_PRESS_MEAS_ATTR_MAX_SCALED_VAL	Maximum scaled value (optional)
ZCL_PRESS_MEAS_ATTR_SCALED_TOL	Scaled tolerance (optional)
ZCL_PRESS_MEAS_ATTR_SCALE	Scale (optional)

5.34 Price cluster

```
#include "zcl/se/zcl.price.h"
```

5.34.1 Price cluster description

Table 529. Price cluster PICS code description

PICS Code	Name	Supported	Notes
SEPR.S	Server	True	-
SEPR.C	Client	True	-
SEPR.S.Afffd	`ClusterRevision` server attribute	True	-
SEPR.S.Afffe	`AttributeReportingStatus` server attribute	False	-
SEPR.S.C00.Rsp	Get Current Price server command	False	-
SEPR.S.C01.Rsp	Get Scheduled Prices server command	False	-
SEPR.S.C02.Rsp	Price Acknowledgment server command	False	-
SEPR.S.C03.Rsp	Get Block Period(s) server command	False	-
SEPR.S.C04.Rsp	`GetConversionFactor` server command	False	-
SEPR.S.C05.Rsp	`GetCalorificValue` server command	False	-

PICS Code	Name	Supported	Notes
SEPR.S.C06.Rsp	GetTariffInformation server command	False	-
SEPR.S.C07.Rsp	GetPriceMatrix server command	False	-
SEPR.S.C08.Rsp	GetBlockThresholds server command	False	-
SEPR.S.C09.Rsp	GetCO2Value server command	False	-
SEPR.S.C0a.Rsp	GetTierLabels server command	False	-
SEPR.S.C0b.Rsp	GetBillingPeriod server command	False	-
SEPR.S.C0c.Rsp	GetConsolidatedBill server command	False	-
SEPR.S.C0d.Rsp	CPPEventResponse server command	False	-
SEPR.S.C0e.Rsp	GetCreditPayment server command	False	-
SEPR.S.C0f.Rsp	GetCurrencyConversion server command	False	-
SEPR.S.C10.Rsp	GetTariffCancellation server command	False	-
SEPR.S.C00.Tx	Publish Price server command	True	-
SEPR.S.C01.Tx	Publish Block Period server command	False	-
SEPR.S.C02.Tx	Publish Conversion Factor server command	False	-
SEPR.S.C03.Tx	Publish Calorific Value server command	False	-
SEPR.S.C04.Tx	PublishTariffInformation server command	True	-
SEPR.S.C05.Tx	PublishPriceMatrix server command	True	-
SEPR.S.C06.Tx	PublishBlockThresholds server command	True	-
SEPR.S.C07.Tx	PublishCO2Value server command	False	-
SEPR.S.C08.Tx	PublishTierLabels server command	False	-
SEPR.S.C09.Tx	PublishBillingPeriod server command	False	-
SEPR.S.C0a.Tx	PublishConsolidatedBill server command	False	-
SEPR.S.C0b.Tx	PublishCPPEvent server command	False	-
SEPR.S.C0c.Tx	PublishCreditPayment server command	False	-
SEPR.S.C0d.Tx	PublishCurrencyConversion server command	False	-
SEPR.S.C0e.Tx	CancelTariff server command	False	-
SEPR.C.Afffd	ClusterRevision client attribute	True	-
SEPR.C.Afffe	AttributeReportingStatus client attribute	False	-
SEPR.C.C00.Rsp	Publish Price client command	False	-
SEPR.C.C01.Rsp	Publish Block Period client command	False	-
SEPR.C.C02.Rsp	Publish Conversion Factor client command	False	-
SEPR.C.C03.Rsp	Publish Calorific Value client command	False	-
SEPR.C.C04.Rsp	PublishTariffInformation client command	False	-
SEPR.C.C05.Rsp	PublishPriceMatrix client command	False	-
SEPR.C.C06.Rsp	PublishBlockThresholds client command	False	-
SEPR.C.C07.Rsp	PublishCO2Value client command	False	-
SEPR.C.C08.Rsp	PublishTierLabels client command	False	-
SEPR.C.C09.Rsp	PublishBillingPeriod client command	False	-
SEPR.C.C0a.Rsp	PublishConsolidatedBill client command	False	-

PICS Code	Name	Supported	Notes
SEPR.C.C0b.Rsp	PublishCPPEvent client command	False	-
SEPR.C.C0c.Rsp	PublishCreditPayment client command	False	-
SEPR.C.C0d.Rsp	PublishCurrencyConversion client command	False	-
SEPR.C.C0e.Rsp	CancelTariff client command	False	-
SEPR.C.C00.Tx	Get Current Price client command	True	-
SEPR.C.C01.Tx	Get Scheduled Prices client command	True	-
SEPR.C.C02.Tx	Price Acknowledgment client command	True	-
SEPR.C.C03.Tx	Get Block Period(s) client command	False	-
SEPR.C.C04.Tx	GetConversionFactor client command	False	-
SEPR.C.C05.Tx	GetCalorificValue client command	False	-
SEPR.C.C06.Tx	GetTariffInformation client command	True	-
SEPR.C.C07.Tx	GetPriceMatrix client command	True	-
SEPR.C.C08.Tx	GetBlockThresholds client command	True	-
SEPR.C.C09.Tx	GetCO2Value client command	False	-
SEPR.C.C0a.Tx	GetTierLabels client command	False	-
SEPR.C.C0b.Tx	GetBillingPeriod client command	False	-
SEPR.C.C0c.Tx	GetConsolidatedBill client command	False	-
SEPR.C.C0d.Tx	CPPEventResponse client command	False	-
SEPR.C.C0e.Tx	GetCreditPayment client command	False	-
SEPR.C.C0f.Tx	GetCurrencyConversion client command	False	-
SEPR.C.C10.Tx	GetTariffCancellation client command	False	-

5.34.2 Price cluster functions

5.34.2.1 *ZbZclPriceClientAlloc*

```
struct ZbZclClusterT * ZbZclPriceClientAlloc(struct ZigBeeT *zb, uint8_t endpoint, struct
ZbZclPriceClientCallbacksT *callbacks, void *arg);
```

Create a new instance of the Price Client cluster

Table 530. **ZbZclPriceClientAlloc parameters**

Parameter	Description
zb	Zigbee® stack instance
endpoint	Endpoint on which to create cluster
callbacks	Structure containing any callback function pointers for this cluster
arg	Pointer to application data that is provided back to the callback function when invoked

Return: Cluster pointer, or NULL if there is an error.

5.34.2.2 *ZbZclPriceClientCommandGetBlockThresholdsReq*

```
enum ZclStatusCodeT ZbZclPriceClientCommandGetBlockThresholdsReq(struct ZbZclClusterT
*cluster, const struct ZbApsAddrT *dst, struct ZbZclPriceClientGetBlockThresholdsT *cmd_req,
void (*callback)(struct ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Get Block Thresholds command

Table 531. ZbZclPriceClientCommandGetBlockThresholdsReq parameters

Parameter	Description
cluster	Cluster instance from which to send this command
dst	Requested destination address
cmd_req	Get Block Thresholds command structure
callback	Callback function that is invoked when the response is received
arg	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.34.2.3 *ZbZclPriceClientCommandGetCurrentPriceReq*

```
enum ZclStatusCodeT ZbZclPriceClientCommandGetCurrentPriceReq(struct ZbZclClusterT*cluster,
const struct ZbApsAddrT *dst, void (*callback)(struct ZbZclCommandRspT*rsp, void *arg), void
*arg);
```

Send a Get Current Price command

Table 532. ZbZclPriceClientCommandGetCurrentPriceReq parameters

Parameter	Description
cluster	Cluster instance from which to send this command
dst	Requested destination address
callback	Callback function that is invoked when the response is received
arg	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.34.2.4 *ZbZclPriceClientCommandGetPriceMatrixReq*

```
enum ZclStatusCodeT ZbZclPriceClientCommandGetPriceMatrixReq(struct ZbZclClusterT*cluster,
const struct ZbApsAddrT *dst, struct ZbZclPriceClientGetPriceMatrixT*cmd_req, void
(*callback)(struct ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Get Price Matrix command

Table 533. ZbZclPriceClientCommandGetPriceMatrixReq parameters

Parameter	Description
cluster	Cluster instance from which to send this command
dst	Requested destination address
cmd_req	Get Scheduled Events command request structure
callback	Callback function that is invoked when the response is received
arg	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.34.2.5 *ZbZclPriceClientCommandGetScheduledPricesReq*

```
enum ZclStatusCodeT ZbZclPriceClientCommandGetScheduledPricesReq(struct ZbZclClusterT
*cluster, const struct ZbApsAddrT *dst, struct ZbZclPriceClientGetScheduledPricesT *cmd_req,
void (*callback)(struct ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Get Scheduled Prices command

Table 534. **ZbZclPriceClientCommandGetScheduledPricesReq parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`cmd_req`	Get Scheduled Prices command structure
`callback`	Callback function that is invoked when the response is received
`arg`	`arg`

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.34.2.6 *ZbZclPriceClientCommandGetTariffInfoReq*

```
enum ZclStatusCodeT ZbZclPriceClientCommandGetTariffInfoReq(struct ZbZclClusterT*cluster,
const struct ZbApsAddrT *dst, struct ZbZclPriceClientGetTariffInfoT*cmd_req, void (*callback)
(struct ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Get Tariff Information command

Table 535. **ZbZclPriceClientCommandGetTariffInfoReq parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`cmd_req`	Get Tariff Information command structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.34.2.7 *ZbZclPriceClientCommandPriceAckReq*

```
enum ZclStatusCodeT ZbZclPriceClientCommandPriceAckReq(struct ZbZclClusterT*cluster, const
struct ZbApsAddrT *dst, struct ZbZclPriceClientPriceAckT *cmd_req, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Price Acknowledgement command

Table 536. **ZbZclPriceClientCommandPriceAckReq parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`cmd_req`	Price Acknowledge command structure
`callback`	Callback function that is invoked when the response is received

Parameter	Description
`arg`	Pointer to application data that is provided back to the callback function when invoked

Pointer to application data that is provided back to the callback function when invoked

5.34.2.8 *ZbZclPriceServerAlloc*

```
struct ZbZclClusterT * ZbZclPriceServerAlloc(struct ZigBeeT *zb, uint8_t endpoint, struct
ZbZclPriceServerCallbacksT *callbacks, void *arg);
```

Create a new instance of the Price Server cluster

Table 537. **ZbZclPriceServerAlloc parameters**

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster
`callbacks`	Structure containing any callback function pointers for this cluster
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: Cluster pointer, or NULL if there is an error.

5.34.2.9 *ZbZclPriceServerPublishPriceInit*

```
void ZbZclPriceServerPublishPriceInit(struct ZbZclPriceServerPublishPriceT *rsp);
```

Initialize Publish Price information

Table 538. **ZbZclPriceServerPublishPriceInit parameters**

Parameter	Description
`rsp`	Publish price command structure

Return: Void

5.34.2.10 *ZbZclPriceServerSendPublishBlockThresholds*

```
enum ZclStatusCodeT ZbZclPriceServerSendPublishBlockThresholds(struct ZbZclClusterT*cluster,
const struct ZbApsAddrT *dst, struct ZbZclPriceServerPublishBlockThresholdsT *notify, void
(*callback)(struct ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Publish Block Thresholds command

Table 539. **ZbZclPriceServerSendPublishBlockThresholds parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Destination address for command, including sequence number and tx options
`notify`	Publish Block Thresholds command structure
`callback`	Callback function that is invoked later when a response is received, if one is expected. If broadcasting, then this must be set to NULL since no response is expected
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.34.2.11 ZbZclPriceServerSendPublishMatrix

```
enum ZclStatusCodeT ZbZclPriceServerSendPublishMatrix(struct ZbZclClusterT*cluster, const
struct ZbApsAddrT *dst, struct ZbZclPriceServerPublishPriceMatrixT*notify, void (*callback)
(struct ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Publish Price Matrix command

Table 540. ZbZclPriceServerSendPublishMatrix parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`notify`	Publish Price Matrix command structure
`callback`	Callback function that is invoked when response is received, if one is expected. If broadcasting, then this must be set to NULL since no response is expected
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.34.2.12 ZbZclPriceServerSendPublishPrice

```
enum ZclStatusCodeT ZbZclPriceServerSendPublishPrice(struct ZbZclClusterT *cluster, const
struct ZbApsAddrT *dst, struct ZbZclPriceServerPublishPriceT *notify, void (*callback)
(struct ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Publish Price command

Table 541. ZbZclPriceServerSendPublishPrice parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Destination address for command, including sequence number and Tx options
`notify`	Publish Price command structure
`callback`	Callback function that is invoked when response is received, if one is expected. If broadcasting, then this must be set to NULL since no response is expected
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.34.2.13 ZbZclPriceServerSendPublishTariffInfo

```
enum ZclStatusCodeT ZbZclPriceServerSendPublishTariffInfo(struct ZbZclClusterT*cluster,
const struct ZbApsAddrT *dst, struct ZbZclPriceServerPublishTariffInfoT*notify, void
(*callback)(struct ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Publish Tariff Information command

Table 542. ZbZclPriceServerSendPublishTariffInfo parameters

Parameter	Description
`dst`	Requested destination address
`notify`	Publish Tariff Info command structure
`callback`	Callback function that is invoked when response is received, if one is expected. If broadcasting, then this is set to NULL since no response is expected

Parameter	Description
arg	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other ZclStatusCodeT value on error.

5.34.3 Price cluster enumerations

5.34.3.1 *ZbZclPriceSvrAttrT*

Table 543. Price server attribute IDs

Attribute	Description
ZCL_PRICE_SVR_ATTR_TIER1_LABEL	Tier1PriceLabel (optional) ZCL_PRICE_SVR_ATTR_TIERN_LABEL(1) For all the tiers, use the ZCL_PRICE_SVR_ATTR_TIERN_LABEL(tier) macro.
ZCL_PRICE_SVR_ATTR_BLOCK1_THRESHOLD	Block1Threshold (optional) ZCL_PRICE_SVR_ATTR_BLOCKN_THRESHOLD(1) For all the blocks, use the ZCL_PRICE_SVR_ATTR_BLOCKN_THRESHOLD (block) macro.
ZCL_PRICE_SVR_ATTR_START_OF_BLOCK_PERIOD	StartofBlockPeriod (optional)
ZCL_PRICE_SVR_ATTR_BLOCK_PERIOD_DURATION	BlockPeriodDuration (optional)
ZCL_PRICE_SVR_ATTR_COMMODITY_TYPE	CommodityType (optional)
ZCL_PRICE_SVR_ATTR_STANDING_CHARGE	StandingCharge (optional)
ZCL_PRICE_SVR_ATTR_NO_TIER_BLOCK1_PRICE	NoTierBlock1Price (optional) ZCL_PRICE_SVR_ATTR_NO_TIER_BLOCKN_PRICE(1) For all the blocks, use the ZCL_PRICE_SVR_ATTR_NO_TIER_BLOCKN_PRICE(block) macro.
ZCL_PRICE_SVR_ATTR_TIER1_BLOCK1_PRICE	Tier1Block1Price (optional) ZCL_PRICE_SVR_ATTR_TIERN_BLOCKN_PRICE(1,1) For all the tiers and blocks, use the ZCL_PRICE_SVR_ATTR_TIERN_BLOCKN_PRICE(tier, block) macro.
ZCL_PRICE_SVR_ATTR_CBP_START	CurrentBillingPeriodStart (optional)
ZCL_PRICE_SVR_ATTR_CBP_DURATION	CurrentBillingPeriodDuration (optional)
ZCL_PRICE_SVR_ATTR_LBP_START	LastBillingPeriodStart (optional)
ZCL_PRICE_SVR_ATTR_LBP_DURATION	LastBillingPeriodDuration (optional)
ZCL_PRICE_SVR_ATTR_LBP_CONSOLIDATED_BILL	LastBillingPeriodConsolidatedBill (optional)

5.34.4 Price cluster structures

5.34.4.1 *ZbZclPriceClientCallbacksT*

Price Client callbacks configuration

Table 544. ZbZclPriceClientCallbacksT parameters

Parameter	Description
`publish_price` (callback function pointer)	`enum ZclStatusCodeT (*publish_price)(struct ZbZclClusterT *cluster, void *arg,struct ZbZclPriceServerPublishPriceT *price, struct ZbZclAddrInfoT *srcInfo)` Callback to Server, invoked on receipt of Publish Price command.
`optional` (callback function pointer)	`Enum ZclStatusCodeT (*optional)(struct ZbZclClusterT*cluster,struct ZbZclHeaderT *zclHdrPtr,struct ZbApsdeDataIndT *dataIndPtr)` Callback to Server, invoked on receipt of optional Server commands.

5.34.4.2 *ZbZclPriceClientGetBlockThresholdsT*

Get Block Thresholds command structure

Table 545. ZbZclPriceClientGetBlockThresholdsT parameters

Parameter	Description
`uint32_t issuer_tariff_id`	Issuer tariff ID

5.34.4.3 *ZbZclPriceClientGetCurrentPriceT*

Get Current Price command structure

Table 546. ZbZclPriceClientGetCurrentPriceT parameters

Parameter	Description
`uint8_t command_options`	Command options

5.34.4.4 *ZbZclPriceClientGetPriceMatrixT*

Get Price Matrix command structure

Table 547. ZbZclPriceClientGetPriceMatrixT parameters

Parameter	Description
`uint32_t issuer_tariff_id`	Issuer tariff ID

5.34.4.5 *ZbZclPriceClientGetScheduledPricesT*

Get Scheduled Prices command structure

Table 548. ZbZclPriceClientGetScheduledPricesT parameters

Parameter	Description
`uint32_t startTime`	Start time
`uint8_t maxPrices`	Number of events

5.34.4.6 *ZbZclPriceClientGetTariffInfoT*

Get Tariff Information command structure

Table 549. ZbZclPriceClientGetTariffInfoT parameters

Parameter	Description
uint32_t earliest_start_time	Earliest start time
uint32_t min_issuer_event_id	Minimum issuer event ID
uint8_t num_commands	Number of commands
uint8_t tariff_type	Tariff type

5.34.4.7 *ZbZclPriceClientPriceAckT*

Price Acknowledge command structure

Table 550. ZbZclPriceClientPriceAckT parameters

Parameter	Description
uint32_t provider_id	Provider ID
uint32_t issuer_event_id	Issuer event ID
uint32_t price_ack_time	Price acknowledge time
uint8_t control	Control. For example ZCL_PRICE_CTRL_MASK_PRICE_ACK

5.34.4.8 *ZbZclPriceServerBlockThreshEntryT*

Block Threshold Entry structure

Table 551. ZbZclPriceServerBlockThreshEntryT parameters

Parameter	Description
uint8_t tier	Tier
uint64_t *block_thresh	Block threshold

5.34.4.9 *ZbZclPriceServerCallbacksT*

Price Server callbacks configuration

Table 552. ZbZclPriceServerCallbacksT parameters

Parameter	Description
get_current_price (callback function pointer)	enum ZclStatusCodeT (*get_current_price)(struct ZbZclClusterT *cluster, void *arg,struct ZbZclPriceClientGetCurrentPriceT *req, struct ZbZclAddrInfoT *srcInfo) Callback to application, invoked on receipt of Get Current command. Application must return ZCL_STATUS_SUCCESS to include in the default response. The application may call ZbZclPriceServerSendPublishPrice to send a PublishPrice command.
price_ack (callback function pointer)	Enum ZclStatusCodeT (*price_ack)(struct ZbZclClusterT*cluster, void *arg,struct ZbZclPriceClientPriceAckT*req, struct ZbZclAddrInfoT *srcInfo) Callback to application, invoked on receipt of Price Acknowledgement command.
get_tariff_info (callback function pointer)	enum ZclStatusCodeT (*get_tariff_info)(struct ZbZclClusterT *cluster, void *arg,struct ZbZclPriceClientGetTariffInfoT *req, struct ZbZclAddrInfoT *srcInfo)

Parameter	Description
	Callback to application, invoked on receipt of `GetTariffInformation` command. Application must return ZCL_STATUS_SUCCESS to include in the default response. The application may call ZbZclPriceServerSendPublishTariffInfo to send a `PublishTariffInfo` command.
`get_price_matrix` (callback function pointer)	`enum ZclStatusCodeT (*get_price_matrix)(struct ZbZclClusterT *cluster, void *arg,struct ZbZclPriceClientGetPriceMatrixT *req, struct ZbZclAddrInfoT *srcInfo)` Callback to application, invoked on receipt of Get Price Matrix command. Application must return ZCL_STATUS_SUCCESS to include in the Default Response. The application may call ZbZclPriceServerSendPublishMatrix to send a `PublishMatrix` command.
`get_block_thresholds` (callback function pointer)	`enum ZclStatusCodeT (*get_block_thresholds)(struct ZbZclClusterT *cluster, void *arg,struct ZbZclPriceClientGetBlockThresholdsT *req, struct ZbZclAddrInfoT *srcInfo)` Callback to application, invoked on receipt of Get Block Thresholds command. Application must return ZCL_STATUS_SUCCESS to include in the Default Response. The application may call ZbZclPriceServerSendPublishBlockThresholdstosenda PublishBlockThresholds command.
`optional` (callback function pointer)	`Enum ZclStatusCodeT (*optional)(struct ZbZclClusterT *cluster, struct ZbZclHeaderT *zclHdrPtr,struct ZbApsdeDataIndT *dataIndPtr)` Callback to application, invoked on receipt of optional Client commands.

5.34.4.10 *ZbZclPriceServerPriceMatrixEntryT*

Price Matrix Entry structure

Table 553. **ZbZclPriceServerPriceMatrixEntryT parameters**

Parameter	Description
`uint8_t tier_block_id`	Tier/block ID
`uint32_t price`	Price

5.34.4.11 *ZbZclPriceServerPublishBlockThresholdsT*

Publish Block Thresholds command structure

Table 554. **ZbZclPriceServerPublishBlockThresholdsT parameters**

Parameter	Description
`uint32_t provider_id`	Provider ID
`uint32_t issuer_event_id`	Issuer event ID
`uint32_t start_time`	Start time - UTC time
`uint32_t issuer_tariff_id`	Issuer tariff ID
`uint8_t command_index`	Command index
`uint8_t total_number_commands`	Total number of commands
`uint8_t sub_payload_control`	Sub-payload control
`uint8_t num_sub_paylad`	Number of items in price matrix sub-payload
`struct ZbZclPriceServerBlockThreshEntryT sub_payload`	Block threshold sub-payload

5.34.4.12 *ZbZclPriceServerPublishPriceMatrixT*

Publish Price Matrix command structure

Table 555. ZbZclPriceServerPublishPriceMatrixT parameters

Parameter	Description
`uint32_t provider_id`	Provider ID
`uint32_t issuer_event_id`	Issuer event ID
`uint32_t start_time`	Start time - UTC time
`uint32_t issuer_tariff_id`	Issuer tariff ID
`uint8_t command_index`	Command index
`uint8_t total_number_commands`	Total number of commands
`uint8_t sub_payload_control`	Sub-payload control
`uint8_t num_sub_paylad`	Number of items in price matrix sub-payload
`struct ZbZclPriceServerPriceMatrixEntryT sub_payload`	Price matrix sub-payload

5.34.4.13 *ZbZclPriceServerPublishPriceT*

Publish Price command structure

Table 556. ZbZclPriceServerPublishPriceT parameters

Parameter	Description
`uint32_t provider_id`	Provider ID
`char rate_lable`	Rate label array
`uint32_t issuer_event_id`	Issuer event ID
`uint32_t current_Time`	Current time - UTC time
`uint8_t unit_of_measure`	Unit of measure
`uint16_t currency`	Currency
`uint8_t trailing_digit_and_price_tier`	Price trailing digit and price tier
`uint8_t num_price_tiers`	Number of price tiers and register tier
`uint32_t start_time`	Start time - UTC time
`uint16_t duration`	Duration in minutes
`uint32_t price`	Price
`uint8_t price_ratio`	Price ratio (optional)
`uint32_t generation_price`	Generation price (optional)
`uint8_t generation_price_ratio`	Generation price ratio (optional)
`uint32_t alternate_cost_delivered`	Alternate cost delivered (optional)
`uint8_t alternate_cost_units`	Alternate cost unit (optional)
`uint8_t alternate_cost_trail_digits`	Alternate cost trailing digit (optional)
`uint8_t num_block_thresholds`	Number of block thresholds (optional)
`uint8_t price_control`	Price control (optional)
`uint8_t number_of_gen_tiers`	Number of generation tiers (optional)
`uint8_t gen_tier`	Generation tier (optional)
`uint8_t ext_number_price_tiers`	Extended number of price tiers (optional)
`uint8_t ext_price_tier`	Extended price tier (optional)

Parameter	Description
`uint8_t ext_register_tier`	Extended register tier (optional)

5.34.4.14 *ZbZclPriceServerPublishTariffInfoT*

Publish Tariff Info command structure

Table 557. ZbZclPriceServerPublishTariffInfoT parameters

Parameter	Description
`uint32_t provider_id`	Provider ID
`uint32_t issuer_event_id`	Issuer event ID
`uint32_t issuer_tariff_id`	Issuer tariff ID
`uint32_t start_time`	Start time - UTC time
`uint8_t tariff_type_and_charging_scheme`	Tariff Type/Charging Scheme
`char tariff_label`	Tariff label array
`uint8_t number_of_price_tiers_in_use`	Number of price tiers in use
`uint8_t number_of_block_thresh_in_use`	Number of Block Thresholds in Use
`uint8_t unit_of_measure`	Unit of measure
`uint16_t currency`	Currency
`uint8_t price_trailing_digit`	Price Trailing Digit
`uint32_t standing_charge`	Standing charge
`uint8_t tier_block_mode`	`TierBlockMode`
`uint32_t block_thresh_multiplier`	Block threshold multiplier - 24-bit
`uint32_t block_thresh_divisor`	Block threshold divisor - 24-bit

5.35 Pump configuration and control cluster

```
#include "zcl/general/zcl.pump.h"
```

5.35.1 Pump configuration and control cluster description

Table 558. Pump configuration and control cluster PICS code description

PICS Code	Name	Supported	Notes
PCC.S	Server	True	-
PCC.C	Client	True	-
PCC.S.A0000	`MaxPressure` server attribute	True	-
PCC.S.A0001	`MaxSpeed` server attribute	True	-
PCC.S.A0002	`MaxFlow` server attribute	True	-
PCC.S.A0010.Report.Tx	`PumpStatus` server attribute reports	False	-
PCC.S.A0011	`EffectiveOperationMode` server attribute	True	-
PCC.S.A0012	`EffectiveControlMode` server attribute	True	-
PCC.S.A0013	`Capacity` server attribute	True	-
PCC.S.A0013.Report.Tx	`Capacity` server attribute reports	True	-

PICS Code	Name	Supported	Notes
PCC.S.A0020	`OperationMode` server attribute	True	-
PCC.S.Afffd	`ClusterRevision` server attribute	True	-
PCC.S.Afffe	`AttributeReportingStatus` server attribute	False	-
PCC.C.A0010.Report.Rsp	`PumpStatus` client attribute reports	False	-
PCC.S.A0013.Report.Rsp	Capacity server attribute reports	False	-
PCC.C.Afffd	`ClusterRevision` client attribute	True	-
PCC.C.Afffe	`AttributeReportingStatus` client attribute	False	-

5.35.2 Pump configuration and control cluster functions

5.35.2.1 *ZbZclPumpClientAlloc*

```
struct ZbZclClusterT * ZbZclPumpClientAlloc(struct ZigBeeT *zb, uint8_t endpoint);
```

Instantiate a new instance of the Pump Configuration and Control client cluster

Table 559. **ZbZclPumpClientAlloc parameters**

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster

Return: Cluster pointer, or NULL if there is an error.

5.35.2.2 *ZbZclPumpServerAlloc*

```
struct ZbZclClusterT * ZbZclPumpServerAlloc(struct ZigBeeT *zb, uint8_t endpoint);
```

Instantiate a new instance of the Pump Configuration and Control server cluster

Table 560. **ZbZclPumpServerAlloc parameters**

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster

Return: Cluster pointer, or NULL if there is an error.

5.35.3 Pump configuration and control cluster enumerations

5.35.3.1 *ZbZclPumpServerAttrT*

Table 561. **Pump Configuration and Control Attribute Ids**

Parameter	Description
ZCL_PUMP_SVR_ATTR_MAX_PRESSURE	`MaxPressure`
ZCL_PUMP_SVR_ATTR_MAX_SPEED	`MaxSpeed`
ZCL_PUMP_SVR_ATTR_MAX_FLOW	`MaxFlow`
ZCL_PUMP_SVR_ATTR_MIN_CONST_PRESSURE	`MinConstPressure`

Parameter	Description
ZCL_PUMP_SVR_ATTR_MAX_CONST_PRESSURE	MaxConstPressure
ZCL_PUMP_SVR_ATTR_MIN_COMP_PRESSURE	MinCompPressure
ZCL_PUMP_SVR_ATTR_MAX_COMP_PRESSURE	MaxCompPressure
ZCL_PUMP_SVR_ATTR_MIN_CONST_SPEED	MinConstSpeed
ZCL_PUMP_SVR_ATTR_MAX_CONST_SPEED	MaxConstSpeed
ZCL_PUMP_SVR_ATTR_MIN_CONST_FLOW	MinConstFlow
ZCL_PUMP_SVR_ATTR_MAX_CONST_FLOW	MaxConstFlow
ZCL_PUMP_SVR_ATTR_MIN_CONST_TEMP	MinConstTemp
ZCL_PUMP_SVR_ATTR_MAX_CONST_TEMP	MaxConstTemp
ZCL_PUMP_SVR_ATTR_PUMP_STATUS	PumpStatus
ZCL_PUMP_SVR_ATTR_EFF_OP_MODE	EffectiveOperationMode
ZCL_PUMP_SVR_ATTR_EFF_CTRL_MODE	EffectiveControlMode
ZCL_PUMP_SVR_ATTR_CAPACITY	Capacity
ZCL_PUMP_SVR_ATTR_SPEED	Speed
ZCL_PUMP_SVR_ATTR_RUNNING_HOURS	LifetimeRunningHours
ZCL_PUMP_SVR_ATTR_POWER	Power
ZCL_PUMP_SVR_ATTR_ENERGY_CONSUMED	LifetimeEnergyConsumed
ZCL_PUMP_SVR_ATTR_OP_MODE	OperationMode
ZCL_PUMP_SVR_ATTR_CTRL_MODE	ControlMode
ZCL_PUMP_SVR_ATTR_ALARM_MASK	AlarmMask

5.36 RSSI location cluster

```
#include "zcl/general/zcl.rssi.loc.h"
```

5.36.1 RSSI location cluster description

Table 562. RSSI location cluster PICS code description

PICS Code	Name	Supported	Notes
RSSI.S	Server	True	-
RSSI.C	Client	True	-
RSSI.S.A0000	LocationType server attribute	True	-
RSSI.S.A0001	LocationMethod server attribute	True	-
RSSI.S.A0002	LocationAge server attribute	True	Optional
RSSI.S.A0003	QualityMeasure server attribute	True	Optional
RSSI.S.A0004	NumberOfDevices server attribute	True	Optional
RSSI.S.A0010	Coordinate1 server attribute	True	-
RSSI.S.A0011	Coordinate2 server attribute	True	-
RSSI.S.A0012	Coordinate3 server attribute	True	Optional
RSSI.S.A0013	Power server attribute	True	-
RSSI.S.A0014	PathLossExponent server attribute	True	-

PICS Code	Name	Supported	Notes
RSSI.S.A0015	`ReportingPeriod` server attribute	True	Optional
RSSI.S.A0016	`CalculationPeriod` server attribute	True	Optional
RSSI.S.A0017	`NumberRSSIMeasurements` server attribute	True	-
RSSI.S.Afffd	`ClusterRevision` server attribute	True	-
RSSI.S.Affe	`AttributeReportingStatus` server attribute	False	-
RSSI.S.C00.Rsp	Set Absolute Location server command	True	-
RSSI.S.C01.Rsp	Set Device Configuration server command	True	-
RSSI.S.C02.Rsp	Get Device Configuration server command	True	-
RSSI.S.C03.Rsp	Get Location Data server command	True	-
RSSI.S.C04.Rsp	RSSI Response server command	True	-
RSSI.S.C05.Rsp	Send Pings server command	True	-
RSSI.S.C06.Rsp	Anchor Node Announce server command	True	-
RSSI.S.C00.Tx	Device configuration response server command	True	-
RSSI.S.C01.Tx	Location data response server command	True	-
RSSI.S.C02.Tx	Location data notification server command	True	-
RSSI.S.C03.Tx	Compact location data notification server command	True	-
RSSI.S.C04.Tx	RSSI Ping server command	True	-
RSSI.S.C05.Tx	RSSI Request server command	True	-
RSSI.S.C06.Tx	Report RSSI Measurements server command	True	-
RSSI.S.C07.Tx	Request Own Location server command	True	-
RSSI.C.Afffd	`ClusterRevision` client attribute	True	-
RSSI.C.Affe	`AttributeReportingStatus` client attribute	False	-
RSSI.C.C00.Rsp	Device configuration response client command	True	-
RSSI.C.C01.Rsp	Location data response client command	True	-
RSSI.C.C02.Rsp	Location data notification client command	True	-
RSSI.C.C03.Rsp	Compact location data notification client command	True	-
RSSI.C.C04.Rsp	RSSI Ping client command	True	-
RSSI.C.C05.Rsp	RSSI Request client command	True	-
RSSI.C.C06.Rsp	Report RSSI Measurements client command	True	-
RSSI.C.C07.Rsp	Request Own Location client command	True	-
RSSI.C.C00.Tx	Set Absolute Location client command	True	-
RSSI.C.C01.Tx	Set Device Configuration client command	True	-
RSSI.C.C02.Tx	Get Device Configuration client command	True	-
RSSI.C.C03.Tx	Get Location Data client command	True	-
RSSI.C.C04.Tx	RSSI Response client command	True	-
RSSI.C.C05.Tx	Send Pings client command	True	-
RSSI.C.C06.Tx	Anchor Node Announce client command	True	-

5.36.2 RSSI location cluster functions

5.36.2.1 *ZbZclRssiLocClientAlloc*

```
struct ZbZclClusterT * ZbZclRssiLocClientAlloc(struct ZigBeeT *zb, uint8_t endpoint, struct
zcl_rssi_loc_client_callbacks_t *callbacks, void *arg);
```

Create a new instance of the RSSI Location Client cluster

Table 563. ZbZclRssiLocClientAlloc parameters

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster
`callbacks`	Structure containing any callback function pointers for this cluster
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: Cluster pointer, or NULL if there is an error.

5.36.2.2 *ZbZclRssiLocClientAnchorNodeAnnc*

```
enum ZclStatusCodeT ZbZclRssiLocClientAnchorNodeAnnc(struct ZbZclClusterT *cluster, const
struct ZbApsAddrT *dst, struct rssi_loc_anchor_node_annc *anchor_node_annc, void (*callback)
(struct ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Anchor Node Announce command

Table 564. ZbZclRssiLocClientAnchorNodeAnnc parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`anchor_node_annc`	Anchor Node Announce command structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.36.2.3 *ZbZclRssiLocClientGetDevConfig*

```
enum ZclStatusCodeT ZbZclRssiLocClientGetDevConfig(struct ZbZclClusterT *cluster, const
struct ZbApsAddrT *dst, struct rssi_loc_get_dev_config *get_dev_config, void (*callback)
(struct ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Get Device Configuration command

Table 565. ZbZclRssiLocClientGetDevConfig parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`get_dev_config`	Get Device Configuration command structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.36.2.4 *ZbZclRssiLocClientGetLocData*

```
enum ZclStatusCodeT ZbZclRssiLocClientGetLocData(struct ZbZclClusterT *cluster, const struct
ZbApsAddrT *dst, struct rssi_loc_get_loc_data *get_loc_data, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Get Location Data command

Table 566. **ZbZclRssiLocClientGetLocData parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`get_loc_data`	Get Location Data command structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.36.2.5 *ZbZclRssiLocClientSendPings*

```
enum ZclStatusCodeT ZbZclRssiLocClientSendPings(struct ZbZclClusterT *cluster, const
struct ZbApsAddrT *dst, struct rssi_loc_send_pings *send_pings, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Send Pings command

Table 567. **ZbZclRssiLocClientSendPings parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`send_pings`	Send Pings command structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.36.2.6 *ZbZclRssiLocClientSendRssiRsp*

```
enum ZclStatusCodeT ZbZclRssiLocClientSendRssiRsp(struct ZbZclClusterT *clusterPtr, struct
ZbZclAddrInfoT *dst_info, struct rssi_loc_rssi_rsp *rsp);
```

Send a RSSI Response command

Table 568. **ZbZclRssiLocClientSendRssiRsp parameters**

Parameter	Description
`clusterPtr`	Cluster instance from which to send this command
`dst_info`	Destination address for request
`rsp`	RSSI Response structure

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.36.2.7 *ZbZclRssiLocClientSetAbsLocation*

```
enum ZclStatusCodeT ZbZclRssiLocClientSetAbsLocation(struct ZbZclClusterT *cluster, const
struct ZbApsAddrT *dst, struct rssi_loc_set_abs_loc *set_abs_loc, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Set Absolute Location command

Table 569. ZbZclRssiLocClientSetAbsLocation parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`set_abs_loc`	Set Absolute Location command structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.36.2.8 *ZbZclRssiLocClientSetDevConfig*

```
enum ZclStatusCodeT ZbZclRssiLocClientSetDevConfig(struct ZbZclClusterT *cluster, const
struct ZbApsAddrT *dst, struct rssi_loc_set_dev_config *set_dev_config, void (*callback)
(struct ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Set Device Configuration command

Table 570. ZbZclRssiLocClientSetDevConfig parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`set_dev_config`	Set Device Config command structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.36.2.9 *ZbZclRssiLocServerAlloc*

```
struct ZbZclClusterT * ZbZclRssiLocServerAlloc(struct ZigBeeT *zb, uint8_t endpoint, struct
zcl_rssi_loc_server_callbacks_t *callbacks, void *arg);
```

Create a new instance of the RSSI Location Server cluster

Table 571. ZbZclRssiLocServerAlloc parameters

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster
`callbacks`	Structure containing any callback function pointers for this cluster

Parameter	Description
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: Cluster pointer, or NULL if there is an error.

5.36.2.10 *ZbZclRssiLocServerCompDataNotif*

```
enum ZclStatusCodeT ZbZclRssiLocServerCompDataNotif(struct ZbZclClusterT *cluster, const
struct ZbApsAddrT *dst, void (*callback)(struct ZbZclCommandRspT *rsp, void*arg), void *arg);
```

Send a Compact Data Notification command

Table 572. **ZbZclRssiLocServerCompDataNotif parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.36.2.11 *ZbZclRssiLocServerLocDataNotif*

```
enum ZclStatusCodeT ZbZclRssiLocServerLocDataNotif(struct ZbZclClusterT *cluster, const
struct ZbApsAddrT *dst, void (*callback)(struct ZbZclCommandRspT *rsp, void*arg), void *arg);
```

Send a Location Data Notification command

Table 573. **ZbZclRssiLocServerLocDataNotif parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.36.2.12 *ZbZclRssiLocServerReportRssi*

```
enum ZclStatusCodeT ZbZclRssiLocServerReportRssi(struct ZbZclClusterT *cluster, const
struct ZbApsAddrT *dst, struct rssi_loc_report_rssi *report_rssi, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Report RSSI command

Table 574. **ZbZclRssiLocServerReportRssi parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`report_rssi`	Report Rssi command structure

Parameter	Description
callback	Callback function that is invoked when the response is received
arg	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.36.2.13 *ZbZclRssiLocServerReqOwnLoc*

```
enum ZclStatusCodeT ZbZclRssiLocServerReqOwnLoc(struct ZbZclClusterT *cluster, const
struct ZbApsAddrT *dst, struct rssi_loc_req_own_loc *req_own_loc, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Request Own Location command

Table 575. **ZbZclRssiLocServerReqOwnLoc parameters**

Parameter	Description
cluster	Cluster instance from which to send this command
dst	Requested destination address
req_own_loc	Request Own Location command structure
callback	Callback function that is invoked when the response is received
arg	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.36.2.14 *ZbZclRssiLocServerRssiPing*

```
enum ZclStatusCodeT ZbZclRssiLocServerRssiPing(struct ZbZclClusterT *cluster, const struct
ZbApsAddrT *dst, void (*callback)(struct ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a RSSI Ping command

Table 576. **ZbZclRssiLocServerRssiPing parameters**

Parameter	Description
cluster	Cluster instance from which to send this command
dst	Requested destination address
callback	Callback function that is invoked when the response is received
arg	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.36.2.15 *ZbZclRssiLocServerRssiReq*

```
enum ZclStatusCodeT ZbZclRssiLocServerRssiReq(struct ZbZclClusterT *cluster, const struct
ZbApsAddrT *dst, void (*callback)(struct ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a RSSI Request command

Table 577. **ZbZclRssiLocServerRssiReq parameters**

Parameter	Description
cluster	Cluster instance from which to send this command

Parameter	Description
`dst`	Requested destination address
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.36.2.16 *ZbZclRssiLocServerSendDevConfigRsp*

```
enum ZclStatusCodeT ZbZclRssiLocServerSendDevConfigRsp(struct ZbZclClusterT*clusterPtr,
struct ZbZclAddrInfoT *dst_info, struct rssi_loc_dev_config_rsp *rsp);
```

Send a Device Configuration Response command

Table 578. **ZbZclRssiLocServerSendDevConfigRsp parameters**

Parameter	Description
`clusterPtr`	Cluster instance from which to send this command
`dst_info`	Destination address for request
`rsp`	Send Device Configuration response structure

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.36.2.17 *ZbZclRssiLocServerSendLocDataRsp*

```
enum ZclStatusCodeT ZbZclRssiLocServerSendLocDataRsp(struct ZbZclClusterT*clusterPtr, struct
ZbZclAddrInfoT *dst_info, struct rssi_loc_loc_data_rsp *rsp);
```

Send a Location Data Response command

Table 579. **ZbZclRssiLocServerSendLocDataRsp parameters**

Parameter	Description
`clusterPtr`	Cluster instance from which to send this command
`dst_info`	Destination address for request
`rsp`	Send Location Data response structure

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.36.3 RSSI location cluster enumerations

5.36.3.1 *ZbZclRssiLocSvrAttrT*

Table 580. **RSSI location attribute IDs**

Attributes	Description
ZCL_RSSI_LOC_SVR_ATTR_LOCATION_TYPE	Location type
ZCL_RSSI_LOC_SVR_ATTR_LOCATION_METHOD	Location method
ZCL_RSSI_LOC_SVR_ATTR_LOCATION_AGE	Location age (optional)
ZCL_RSSI_LOC_SVR_ATTR_QUALITY_MEAS	Quality measure (optional)
ZCL_RSSI_LOC_SVR_ATTR_NUM_DEVICES	Number of devices (optional)

Attributes	Description
ZCL_RSSI_LOC_SVR_ATTR_COORD1	Coordinate 1
ZCL_RSSI_LOC_SVR_ATTR_COORD2	Coordinate 2
ZCL_RSSI_LOC_SVR_ATTR_COORD3	Coordinate 3 (optional)
ZCL_RSSI_LOC_SVR_ATTR_POWER	Power
ZCL_RSSI_LOC_SVR_ATTR_PATH_LOSS_EXP	Path loss exponent
ZCL_RSSI_LOC_SVR_ATTR_REPORT_PERIOD	Reporting period (optional)
ZCL_RSSI_LOC_SVR_ATTR_CALC_PERIOD	Calculation period (optional)
ZCL_RSSI_LOC_SVR_ATTR_NUM_RSSI_MEAS	Number RSSI measurements

5.36.4 RSSI location cluster structures

5.36.4.1 *rssi_loc_anchor_node_annc*

Anchor Node Announce command structure

Table 581. **rssi_loc_anchor_node_annc parameters**

Parameter	Description
`uint64_t addr`	Anchor Node IEEE Address
`int16_t x`	X
`int16_t y`	Y
`int16_t z`	Z

5.36.4.2 *rssi_loc_comp_data_notif*

Compact Data Notification command structure

Table 582. **rssi_loc_comp_data_notif parameters**

Parameter	Description
`uint8_t loc_type`	Location Type
`int16_t coord1`	Coordinate 1
`int16_t coord2`	Coordinate 2
`int16_t coord3`	Coordinate 3
`uint8_t quality_meas`	Quality Measure
`uint16_t loc_age`	Location Age

5.36.4.3 *rssi_loc_dev_config_rsp*

Device Configuration response structure

Table 583. **rssi_loc_dev_config_rsp parameters**

Parameter	Description
`enum ZclStatusCodeT status`	Status
`int16_t power`	Power
`uint16_t path_loss_exp`	Path loss exponent
`uint16_t calc_period`	Calculation period

Parameter	Description
`uint8_t num_rssi_meas`	Number RSSI measurements
`uint16_t report_period`	Reporting period

5.36.4.4 *rssi_loc_get_dev_config*

Get Device Configuration command structure

Table 584. rssi_loc_get_dev_config parameters

Parameter	Description
`uint64_t target_addr`	Target address

5.36.4.5 *rssi_loc_get_loc_data*

Get Location Data command structure

Table 585. rssi_loc_get_loc_data parameters

Parameter	Description
`uint8_t bitmap`	Bitmap
`uint8_t num_responses`	Number responses
`uint64_t target_addr`	Target address

5.36.4.6 *rssi_loc_loc_data_notif*

Location Data Notification command structure

Table 586. rssi_loc_loc_data_notif parameters

Parameter	Description
`uint8_t loc_type`	Location type
`int16_t coord1`	Coordinate 1
`int16_t coord2`	Coordinate 2
`int16_t coord3`	Coordinate 3
`int16_t power`	Power
`uint16_t path_loss_exp`	Path loss exponent
`uint8_t loc_method`	Location method
`uint8_t quality_meas`	Quality measure
`uint16_t loc_age`	Location age

5.36.4.7 *rssi_loc_loc_data_rsp*

Location Data response structure

Table 587. rssi_loc_loc_data_rsp parameters

Parameter	Description
`enum ZclStatusCodeT status`	Status
`uint8_t loc_type`	Location Type
`int16_t coord1`	Coordinate 1

Parameter	Description
`int16_t coord2`	Coordinate 2
`int16_t coord3`	Coordinate 3
`int16_t power`	Power
`uint16_t path_loss_exp`	Path loss exponent
`uint8_t loc_method`	Location method
`uint8_t quality_meas`	Quality measure
`uint16_t loc_age`	Location age

5.36.4.8 *rssi_loc_neighbour_info*

Neighbor Info structure

Table 588. **rssi_loc_neighbour_info parameters**

Parameter	Description
`uint64_t neighbour`	Neighbor
`int16_t x`	X
`int16_t y`	Y
`int16_t z`	Z
`int8_t rssi`	RSSI
`uint8_t num_rssi_meas`	Number RSSI measurements

5.36.4.9 *rssi_loc_report_rssi*

Report RSSI command structure

Table 589. **rssi_loc_report_rssi parameters**

Parameter	Description
`uint64_t measuring_dev`	Measuring device
`uint8_t n_neighbours`	N neighbors
`struct rssi_loc_neighbour_infoneighbours_info`	Neighbors info

5.36.4.10 *rssi_loc_req_own_loc*

Request Own Location command structure

Table 590. **rssi_loc_req_own_loc parameters**

Parameter	Description
`uint64_t addr`	IEEE address of the blind node

5.36.4.11 *rssi_loc_rssi_ping*

RSSI Ping command structure

Table 591. **rssi_loc_rssi_ping parameters**

Parameter	Description
`uint8_t loc_type`	Location type

5.36.4.12 *rssi_loc_rssi_req*

RSSI Request structure

Table 592. rssi_loc_rssi_req parameters

Parameter	Description
`uint8_t rssi`	RSSI

5.36.4.13 *rssi_loc_rssi_rsp*

RSSI response structure

Table 593. rssi_loc_rssi_rsp parameters

Parameter	Description
`uint64_t replying_dev`	Replying device
`int16_t x`	X
`int16_t y`	Y
`int16_t z`	Z
`int8_t rssi`	RSSI
`uint8_t num_rssi_meas`	Number RSSI measurements

5.36.4.14 *rssi_loc_send_pings*

Send Pings command structure

Table 594. rssi_loc_send_pings parameters

Parameter	Description
`uint64_t target_addr`	Target address
`uint8_t num_rssi_meas`	Number RSSI measurements
`uint16_t calc_period`	Calculation period

5.36.4.15 *rssi_loc_set_abs_loc*

Set Absolute Location command structure

Table 595. rssi_loc_set_abs_loc parameters

Parameter	Description
`int16_t coord1`	Coordinate 1
`int16_t coord2`	Coordinate 2
`int16_t coord3`	Coordinate 3
`int16_t power`	Power
`uint16_t path_loss_exp`	Path loss exponent

5.36.4.16 *rssi_loc_set_dev_config*

Set Device Configuration command structure

Table 596. rssi_loc_set_dev_config parameters

Parameter	Description
`int16_t power`	Power
`uint16_t path_loss_exp`	Path loss exponent
`uint16_t calc_period`	Calculation period
`uint8_t num_rssi_meas`	Number RSSI measurements
`uint16_t report_period`	Reporting period

5.36.4.17 *zcl_rssi_loc_client_callbacks_t*

RSSI Location Client callbacks configuration

Table 597. zcl_rssi_loc_client_callbacks_t parameters

Parameter	Description
`loc_data_notif` (callback function pointer)	`enum ZclStatusCodeT (*loc_data_notif)(struct ZbZclClusterT *clusterPtr, struct rssi_loc_loc_data_notif *cmd_req,struct ZbZclAddrInfoT*src_info, void *arg)` Callback to application, invoked on receipt of Location Data Notification command
`comp_data_notif` (callback function pointer)	`enum ZclStatusCodeT (*comp_data_notif)(struct ZbZclClusterT*clusterPtr, struct rssi_loc_comp_data_notif *cmd_req,struct ZbZclAddrInfoT*src_info, void *arg)` Callback to application, invoked on receipt of Compact Data Notification command
`rssi_ping` (callback function pointer)	`Enum ZclStatusCodeT (*rssi_ping)(struct ZbZclClusterT*clusterPtr, struct rssi_loc_rssi_ping*cmd_req,struct ZbZclAddrInfoT *src_info, void *arg)` Callback to application, invoked on receipt of RSSI Ping command

`rssi_req`
(callback function pointer)
	`Enum ZclStatusCodeT (*rssi_req)(struct ZbZclClusterT`
	`*clusterPtr, struct rssi_loc_rssi_req *cmd_req,struct ZbZclAddrInfoT *src_info, void *arg)`
	Callback to application, invoked on receipt of RSSI request.

`report_rssi` (callback function pointer)	`enum ZclStatusCodeT (*report_rssi)(struct ZbZclClusterT*clusterPtr, struct rssi_loc_report_rssi *cmd_req,struct ZbZclAddrInfoT *src_info, void *arg)` Callback to application, invoked on receipt of Report RSSI command
`req_own_loc` (callback function pointer)	`enum ZclStatusCodeT (*req_own_loc)(struct ZbZclClusterT*clusterPtr, struct rssi_loc_req_own_loc *cmd_req,struct ZbZclAddrInfoT *src_info, void *arg)` Callback to application, invoked on receipt of Request Own Location command

5.36.4.18 *zcl_rssi_loc_server_callbacks_t*

RSSI Location Server callbacks configuration

Table 598. zcl_rssi_loc_server_callbacks_t parameters

Parameter	Description
`get_dev_config` (callback function pointer)	`Enum ZclStatusCodeT (*get_dev_config)(struct ZbZclClusterT *clusterPtr, struct rssi_loc_get_dev_config*cmd_req,struct ZbZclAddrInfoT*src_info, void *arg)` Callback to application, invoked on receipt of Get Device Configuration command
`get_loc_data` (callback function pointer)	`enum ZclStatusCodeT (*get_loc_data)(struct ZbZclClusterT *clusterPtr, struct rssi_loc_get_loc_data *cmd_req,struct ZbZclAddrInfoT *src_info, void *arg)` Callback to application, invoked on receipt of Get Location Data command
`send_pings` (callback function pointer)	``Enum ZclStatusCodeT (*send_pings)(struct `ZbZclClusterT *clusterPtr, struct rssi_loc_send_pings *cmd_req,struct ZbZclAddrInfoT *src_info, void *arg)`` Callback to application, invoked on receipt of Send Pings command
`anchor_node_annc` (callback function pointer)	`enum ZclStatusCodeT (*anchor_node_annc)(struct ZbZclClusterT *clusterPtr, struct rssi_loc_anchor_node_annc *cmd_req,struct ZbZclAddrInfoT *src_info, void *arg)` Callback to application, invoked on receipt of Anchor Node Announce command

5.37 Scenes cluster

```
#include "zcl/general/zcl.scenes.h"
```

5.37.1 Scenes cluster description

Table 599. Scenes cluster PICS code description

PICS Code	Name	Supported	Notes
S.S	Server	True	-
S.C	Client	True	-
S.S.A0000	`SceneCount` server attribute	True	-
S.S.A0001	`CurrentScene` server attribute	True	-
S.S.A0002	`CurrentGroup` server attribute	True	-
S.S.A0003	`SceneValid` server attribute	True	-
S.S.A0004	`NameSupport` server attribute	True	-
S.S.A0005	`LastConfiguredBy` server attribute	True	Optional
S.S.Afffd	`ClusterRevision` server attribute	True	-
S.S.Afffe	`AttributeReportingStatus` server attribute	False	-
S.S.C00.Rsp	Add Scene server command	True	-
S.S.C01.Rsp	View Scene server command	True	-
S.S.C02.Rsp	Remove Scene server command	True	-
S.S.C03.Rsp	Remove All Scenes server command	True	-
S.S.C04.Rsp	Store Scene server command	True	-
S.S.C05.Rsp	Recall Scene server command	True	-
S.S.C06.Rsp	Get Scene Membership server command	True	-
S.S.C40.Rsp	Enhanced Add Scene server command	True	-

PICS Code	Name	Supported	Notes
S.S.C41.Rsp	Enhanced View Scene server command	True	-
S.S.C42.Rsp	Copy Scene server command	True	-
S.S.C00.Tx	Add Scene Response server command	True	-
S.S.C01.Tx	View Scene Response server command	True	-
S.S.C02.Tx	Remove Scene Response server command	True	-
S.S.C03.Tx	Remove All Scenes Response server command	True	-
S.S.C04.Tx	Store Scene Response server command	True	-
S.S.C06.Tx	Get Scene Membership Response server command	True	-
S.S.C40.Tx	Enhanced Add Scene Response server command	True	-
S.S.C41.Tx	Enhanced View Scene Response server command	True	-
S.S.C42.Tx	Copy Scene Response server command	True	-
S.C.Afffd	`ClusterRevision` client attribute	True	-
S.C.Afffe	`AttributeReportingStatus` client attribute	False	-
S.C.C00.Rsp	Add Scene Response client command	True	-
S.C.C01.Rsp	View Scene Response client command	True	-
S.C.C02.Rsp	Remove Scene Response client command	True	-
S.C.C03.Rsp	Remove All Scenes Response client command	True	-
S.C.C04.Rsp	Store Scene Response client command	True	-
S.C.C06.Rsp	Get Scene Membership Response client command	True	-
S.C.C40.Rsp	Enhanced Add Scene Response client command	True	-
S.C.C41.Rsp	Enhanced View Scene Response client command	True	-
S.C.C42.Rsp	Copy Scene Response client command	True	-
S.C.C00.Tx	Add Scene client command	True	-
S.C.C01.TX	View Scene client command	True	-
S.C.C02.Tx	Remove Scene client command	True	-
S.C.C03.Tx	Remove All Scenes client command	True	-
S.C.C04.Tx	Store Scene client command	True	-
S.C.C05.Tx	Recall Scene client command	True	-
S.C.C06.Tx	Get Scene Membership client command	True	-
S.C.C40.Tx	Enhanced Add Scene client command	True	-
S.C.C41.Tx	Enhanced View Scene client command	True	-
S.C.C42.Tx	Copy Scene client command	True	-

5.37.2 Scenes cluster functions

5.37.2.1 *ZbZclScenesClientAlloc*

```
struct ZbZclClusterT * ZbZclScenesClientAlloc(struct ZigBeeT *zb, uint8_t endpoint);
```

Create a new instance of the Scenes Client cluster

Table 600. ZbZclScenesClientAlloc parameters

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster

Return: Cluster pointer, or NULL if there is an error.

5.37.2.2 *ZbZclScenesServerAlloc*

```
struct ZbZclClusterT * ZbZclScenesServerAlloc(struct ZigBeeT *zb, uint8_t endpoint, uint8_t
maxScenes);
```

Create a new instance of the Scenes Server cluster

Table 601. ZbZclScenesServerAlloc parameters

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster
`maxScenes`	Maximum number of scenes supported by this cluster

Return: Cluster pointer, or NULL if there is an error.

5.37.2.3 *zcl_scenes_client_add_req*

```
enum ZclStatusCodeT zcl_scenes_client_add_req(struct ZbZclClusterT *cluster, struct
zcl_scenes_add_request_t *add_req, void (*callback)(struct ZbZclCommandRspT*zcl_rsp, void
*arg), void *arg);
```

Send an Add Scene or Enhanced Add Scene command, depending on is Enhanced flag in Add Scene command structure

Table 602. zcl_scenes_client_add_req parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`add_req`	Add Scene command structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.37.2.4 *zcl_scenes_client_add_rsp_parse*

```
enum ZclStatusCodeT zcl_scenes_client_add_rsp_parse(struct zcl_scenes_add_response_t
*add_rsp, struct ZbZclCommandRspT *zcl_rsp);
```

Parse an Add Scene Response command payload into a structure

Table 603. zcl_scenes_client_add_rsp_parse parameters

Parameter	Description
`add_rsp`	Add Scene Response command structure
`zcl_rsp`	Cluster response structure

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.37.2.5 *zcl_scenes_client_copy_req*

```
enum ZclStatusCodeT zcl_scenes_client_copy_req(struct ZbZclClusterT *cluster, struct
zcl_scenes_copy_request_t *copy_req, void (*callback)(struct ZbZclCommandRspT *zcl_rsp, void
*arg), void *arg);
```

Send a Copy Scene command

Table 604. zcl_scenes_client_copy_req parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`copy_req`	Copy Scene command structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.37.2.6 *zcl_scenes_client_copy_rsp_parse*

```
enum ZclStatusCodeT zcl_scenes_client_copy_rsp_parse(struct zcl_scenes_copy_response_t
*copy_rsp, struct ZbZclCommandRspT *zcl_rsp);
```

Parse a Copy Scene Response command payload into a structure

Table 605. zcl_scenes_client_copy_rsp_parse parameters

Parameter	Description
`copy_rsp`	Copy Scene response command structure
`zcl_rsp`	Cluster response structure

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.37.2.7 *zcl_scenes_client_get_membership_req*

```
enum ZclStatusCodeT zcl_scenes_client_get_membership_req(struct ZbZclClusterT*cluster,
struct zcl_scenes_membership_request_t *get_req, void (*callback)(struct ZbZclCommandRspT
*zcl_rsp, void *arg), void *arg);
```

Send a Get Scene Membership command

Table 606. zcl_scenes_client_get_membership_req parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`get_req`	Get Scene Membership command structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.37.2.8 *zcl_scenes_client_get_membership_rsp_parse*

```
enum ZclStatusCodeT zcl_scenes_client_get_membership_rsp_parse(struct
zcl_scenes_membership_response_t *get_rsp, struct ZbZclCommandRspT *zcl_rsp);
```

Parse a Get Scene Membership Response command payload into a structure

Table 607. **zcl_scenes_client_get_membership_rsp_parse parameters**

Parameter	Description
`get_rsp`	Get Scene Membership Response command structure
`zcl_rsp`	Cluster response structure

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.37.2.9 *zcl_scenes_client_recall_req*

```
enum ZclStatusCodeT zcl_scenes_client_recall_req(struct ZbZclClusterT *cluster, struct
zcl_scenes_recall_request_t *recall_req, void (*callback)(struct ZbZclCommandRspT *zcl_rsp,
void *arg), void *arg);
```

Send a Recall Scene command

Table 608. **zcl_scenes_client_recall_req parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`recall_req`	Recall Scene command structure
callback	Callback function that is invoked when an alarm occurs
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.37.2.10 *zcl_scenes_client_recall_rsp_parse*

```
enum ZclStatusCodeT zcl_scenes_client_recall_rsp_parse(struct zcl_scenes_recall_response_t
*recall_rsp, struct ZbZclCommandRspT *zcl_rsp);
```

Parse a Recall Scene Response command payload into a structure

Table 609. **zcl_scenes_client_recall_rsp_parse parameters**

Parameter	Description
`recall_rsp`	Recall Scene response command structure
`zcl_rsp`	Cluster response structure

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.37.2.11 *zcl_scenes_client_remove_all_req*

```
enum ZclStatusCodeT zcl_scenes_client_remove_all_req(struct ZbZclClusterT *cluster, struct
zcl_scenes_remove_all_request_t *remove_req, void (*callback)(struct ZbZclCommandRspT
*zcl_rsp, void *arg), void *arg);
```

Send a Remove All Scenes command

Table 610. zcl_scenes_client_remove_all_req parameters

Parameter	Description
cluster	Cluster instance from which to send this command
remove_req	Remove All Scenes command structure
callback	Callback function that is invoked when the response is received
arg	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.37.2.12 *zcl_scenes_client_remove_all_rsp_parse*

```
enum ZclStatusCodeT zcl_scenes_client_remove_all_rsp_parse(struct
zcl_scenes_remove_all_response_t *remove_rsp, struct ZbZclCommandRspT *zcl_rsp);
```

Parse a Remove All Scenes Response command payload into a structure

Table 611. zcl_scenes_client_remove_all_rsp_parse parameters

Parameter	Description
remove_rsp	Remove All Scenes Response command structure
zcl_rsp	Cluster response structure

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.37.2.13 *zcl_scenes_client_remove_req*

```
enum ZclStatusCodeT zcl_scenes_client_remove_req(struct ZbZclClusterT *cluster, struct
zcl_scenes_remove_request_t *remove_req, void (*callback)(struct ZbZclCommandRspT *zcl_rsp,
void *arg), void *arg);
```

Send a Remove Scene command

Table 612. zcl_scenes_client_remove_req parameters

Parameter	Description
cluster	Cluster instance from which to send this command
remove_req	Remove Scene command structure
callback	Callback function that is invoked when the response is received
arg	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.37.2.14 *zcl_scenes_client_remove_rsp_parse*

```
enum ZclStatusCodeT zcl_scenes_client_remove_rsp_parse(struct zcl_scenes_remove_response_t
*remove_rsp, struct ZbZclCommandRspT *zcl_rsp);
```

Parse a Remove Scene Response command payload into a structure

Table 613. zcl_scenes_client_remove_rsp_parse parameters

Parameter	Description
remove_rsp	Remove Scene Response command structure

Parameter	Description
zcl_rsp	Cluster response structure

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.37.2.15 *zcl_scenes_client_store_req*

```
enum ZclStatusCodeT zcl_scenes_client_store_req(struct ZbZclClusterT *cluster, struct
zcl_scenes_store_request_t *store_req, void (*callback)(struct ZbZclCommandRspT *zcl_rsp,
void *arg), void *arg);
```

Send a Store Scene command

Table 614. zcl_scenes_client_store_req parameters

Parameter	Description
cluster	Cluster instance from which to send this command
store_req	Store Scene command structure
callback	Callback function that is invoked when the response is received
arg	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.37.2.16 *zcl_scenes_client_store_rsp_parse*

```
enum ZclStatusCodeT zcl_scenes_client_store_rsp_parse(struct zcl_scenes_store_response_t
*store_rsp, struct ZbZclCommandRspT *zcl_rsp);
```

Parse a Store Scene Response command payload into a structure

Table 615. zcl_scenes_client_store_rsp_parse parameters

Parameter	Description
store_rsp	Store Scene Response command structure
zcl_rsp	Cluster response structure

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.37.2.17 *zcl_scenes_client_view_req*

```
enum ZclStatusCodeT zcl_scenes_client_view_req(struct ZbZclClusterT *cluster, struct
zcl_scenes_view_request_t *view_req, void (*callback)(struct ZbZclCommandRspT *zcl_rsp, void
*arg), void *arg);
```

Send an View Scene or Enhanced View Scene command, depending on is Enhanced flag in View Scene command structure

Table 616. zcl_scenes_client_view_req parameters

Parameter	Description
cluster	Cluster instance from which to send this command
view_req	View Scene command structure
callback	Callback function that is invoked when the response is received
arg	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.37.2.18 *zcl_scenes_client_view_rsp_parse*

```
enum ZclStatusCodeT zcl_scenes_client_view_rsp_parse(struct zcl_scenes_view_response_t
*view_rsp, struct ZbZclCommandRspT *zcl_rsp);
```

Parse a View Scene Response command payload into a structure

Table 617. zcl_scenes_client_view_rsp_parse parameters

Parameter	Description
`view_rsp`	View Scene Response command structure
`zcl_rsp`	Cluster response structure

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.37.3 Scenes cluster enumerations

5.37.3.1 *ZbZclScenesAttrT*

Table 618. Scenes Attribute IDs

Parameter	Description
ZCL_SCENES_ATTR_SCENE_COUNT	Scene count
ZCL_SCENES_ATTR_CURRENT_SCENE	Current scene
ZCL_SCENES_ATTR_CURRENT_GROUP	Current group
ZCL_SCENES_ATTR_SCENE_VALID	Scene valid
ZCL_SCENES_ATTR_NAME_SUPPORT	Name support
ZCL_SCENES_ATTR_LAST_CONFIGURED_BY	Last configured by (optional)

5.37.4 Scenes cluster structures

5.37.4.1 *zcl_scenes_add_request_t*

Add Scene command structure

Table 619. zcl_scenes_add_request_t parameters

Parameter	Description
`bool isEnhanced`	If true, send an Enhanced Add Scene request
`struct ZbApsAddrT dst`	Destination Address
`uint16_t group_id`	Group ID
`uint8_t sceneId`	Scene ID
`uint16_t transition_time`	Transition time
`const char *sceneName`	Scene Name(string length must be ≤ ZCL_SCENES_NAME_MAX_LENGTH)
`const char *extStrPtr`	Extension field sets as an ASCII hex string in the format as sent in the command payload. As per the ZCL Spec, each set has the format: `{clusterId, length, {extension field set}}` E.g. For the On/Off cluster cluster: "06000101"

5.37.4.2 *zcl_scenes_add_response_t*

Add Scene Response command structure

Table 620. zcl_scenes_add_response_t parameters

Parameter	Description
`uint8_t status`	Status
`uint16_t group_id`	Group ID
`uint8_t sceneId`	Scene ID

5.37.4.3 *zcl_scenes_copy_request_t*

Copy Scene command structure

Table 621. zcl_scenes_copy_request_t parameters

Parameter	Description
`struct ZbApsAddrT dst`	Destination Address
`bool allScenes`	Copy All Scenes - sets bit within Mode parameter to enable Copy All Scenes
`uint16_t groupFrom`	Group identifier from
`uint8_t sceneFrom`	Scene identifier from - only used if `allScenes` is FALSE
`uint16_t groupTo`	Group identifier to
`uint8_t sceneTo`	Scene identifier to - only used if `allScenes` is FALSE

5.37.4.4 *zcl_scenes_copy_response_t*

Copy Scene Response command structure

Table 622. zcl_scenes_copy_response_t parameters

Parameter	Description
`uint8_t status`	Status
`uint16_t groupFrom`	Group identifier from
`uint8_t sceneFrom`	Scene identifier from

5.37.4.5 *zcl_scenes_membership_request_t*

Get Scene Membership command structure

Table 623. zcl_scenes_membership_request_t parameters

Parameter	Description
`struct ZbApsAddrT dst`	Destination Address
`uint16_t group_id`	Group ID

5.37.4.6 *zcl_scenes_membership_response_t*

Get Scene Membership Response command structure

Table 624. zcl_scenes_membership_response_t parameters

Parameter	Description
`uint8_t status`	Status
`uint8_t capacity`	Capacity
`uint16_t group_id`	Group ID
`uint8_t sceneCount`	Scent count
`uint8_t sceneList`	Scene list

5.37.4.7 *zcl_scenes_recall_request_t*

Recall Scene command structure

Table 625. zcl_scenes_recall_request_t parameters

Parameter	Description
`struct ZbApsAddrT dst`	Destination Address
`uint16_t group_id`	Group ID
`uint8_t sceneId`	Scene ID
`uint16_t transition_time`	Transition time - time in 1/10ths of second. ZCL_SCENES_RECALL_TRANSITION_INVALID (0xffff) means invalid, and won't be included. (optional)

5.37.4.8 *zcl_scenes_recall_response_t*

Recall Scene Response command structure

Table 626. zcl_scenes_recall_response_t parameters

Parameter	Description
`uint8_t status`	Status

5.37.4.9 *zcl_scenes_remove_all_request_t*

Remove All Scenes command structure

Table 627. zcl_scenes_remove_all_request_t parameters

Parameter	Description
`struct ZbApsAddrT dst`	Destination Address
`uint16_t group_id`	Group ID

5.37.4.10 *zcl_scenes_remove_all_response_t*

Remove All Scenes Response command structure

Table 628. zcl_scenes_remove_all_response_t parameters

Parameter	Description
`uint8_t status`	Status
`uint16_t group_id`	Group ID

5.37.4.11 *zcl_scenes_remove_request_t*

Remove Scene command structure

Table 629. zcl_scenes_remove_request_t parameters

Parameter	Description
`struct ZbApsAddrT dst`	Destination Address
`uint16_t group_id`	Group ID
`uint8_t sceneId`	Scene ID

5.37.4.12 *zcl_scenes_remove_response_t*

Remove Scene Response command structure

Table 630. zcl_scenes_remove_response_t parameters

Parameter	Description
`uint8_t status`	Status
`uint16_t group_list`	Group list
`uint8_t sceneId`	Scene ID

5.37.4.13 *zcl_scenes_store_request_t*

Store Scene command structure

Table 631. zcl_scenes_store_request_t parameters

Parameter	Description
`struct ZbApsAddrT dst`	Destination Address
`uint16_t group_list`	Group list
`uint8_t sceneId`	Scene ID

5.37.4.14 *zcl_scenes_store_response_t*

Store Scene Response command structure

Table 632. zcl_scenes_store_response_t parameters

Parameter	Description
`uint8_t status`	Status
`uint16_t group_list`	Group list
`uint8_t sceneId`	Scene ID

5.37.4.15 *zcl_scenes_view_request_t*

View Scene command structure

Table 633. zcl_scenes_view_request_t parameters

Parameter	Description
`bool isEnhanced`	If true, send an Enhanced View Scene request
`struct ZbApsAddrT dst`	Destination Address

Parameter	Description
`uint16_t group_list`	Group list
`uint8_t sceneId`	Scene ID

5.37.4.16 *zcl_scenes_view_response_t*

View Scene Response command structure

Table 634. zcl_scenes_view_response_t parameters

Parameter	Description
`uint8_t status`	Status
`uint16_t group_list`	Group list
`uint8_t sceneId`	Scene ID
`uint16_t transition`	Transition time
`char nameStr`	Scene name
`uint8_t extNum`	Number of extension field sets
`struct {uint16_t clusterId`	Cluster ID
`uint8_t length`	Length
`uint8_t field`	Field - ZCL_SCENES_VIEW_EXT_FIELD_MAX_LEN

5.38 Temperature measurement cluster

```
#include "zcl/general/zcl.temp.meas.h"
```

5.38.1 Temperature measurement cluster description

Table 635. Temperature measurement cluster PICS code description

PICS Code	Name	Supported	Notes
TM.S	Server	True	-
TM.C	Client	True	-
TM.S.A0000	`MeasuredValue` server attribute	True	-
TM.S.A0000.Report.Tx	`MeasuredValue` server attribute reports	True	-
TM.S.A0001	`MinMeasuredValue` server attribute	True	-
TM.S.A0002	`MaxMeasuredValue` server attribute	True	-
TM.S.A0003	Tolerance server attribute	True	Optional
TM.S.Afffd	`ClusterRevision` server attribute	True	-
TM.S.Afffe	`AttributeReportingStatus` server attribute	False	-
TM.C.A0000.Report.Rsp	`MeasuredValue` client attribute reports	False	-
TM.C.Afffd	`ClusterRevision` client attribute	True	-
TM.C.Afffe	`AttributeReportingStatus` client attribute	False	-

5.38.2 Temperature measurement cluster functions

5.38.2.1 *ZbZclTempMeasClientAlloc*

```
struct ZbZclClusterT * ZbZclTempMeasClientAlloc(struct ZigBeeT *zb, uint8_t endpoint);
```

Create a new instance of the Temperature Measurement Client cluster

Table 636. ZbZclTempMeasClientAlloc parameters

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster

Return: Cluster pointer, or NULL if there is an error.

5.38.2.2 *ZbZclTempMeasServerAlloc*

```
struct ZbZclClusterT * ZbZclTempMeasServerAlloc(struct ZigBeeT *zb, uint8_t endpoint,
int16_t min, int16_t max, uint16_t tolerance);
```

Create a new instance of the Temperature Measurement Server cluster

Table 637. ZbZclTempMeasServerAlloc parameters

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster
`min`	Minimum value capable of being measured (`MinMeasuredValue`)
`max`	Maximum value capable of being measured (`MaxMeasuredValue`)
`tolerance`	Tolerance

Return: Cluster pointer, or NULL if there is an error.

5.38.3 Temperature measurement cluster enumerations

5.38.3.1 *ZbZclTempMeasSvrAttrT*

Table 638. Temperature measurement server attribute IDs

Parameter	Description
ZCL_TEMP_MEAS_ATTR_MEAS_VAL	`MeasuredValue`
ZCL_TEMP_MEAS_ATTR_MIN_MEAS_VAL	`MinMeasuredValue`
ZCL_TEMP_MEAS_ATTR_MAX_MEAS_VAL	`MaxMeasuredValue`
ZCL_TEMP_MEAS_ATTR_TOLERANCE	Tolerance (optional)

5.39 Thermostat cluster

```
#include "zcl/general/zcl.therm.h"
```

5.39.1 Thermostat cluster description

Table 639. Thermostat cluster PICS code description

PICS Code	Name	Supported	Notes
TSTAT.S	Server	True	-
TSTAT.C	Client	True	-
TSTAT.S.A0000	`LocalTemperature` server attribute	True	-
TSTAT.S.A0000.Report.Tx	`LocalTemperature` server attribute reports	True	-
TSTAT.S.A0001	`OutdoorTemperature` server attribute	True	Optional
TSTAT.S.A0011	`OccupiedCoolingSetpoint` server attribute	True	-
TSTAT.S.A0011.Scene	`OccupiedCoolingSetpoint` server attribute scenes	True	-
TSTAT.S.A0012	`OccupiedHeatingSetpoint` server attribute	True	-
TSTAT.S.A0012.Scene	`OccupiedHeatingSetpoint` server attribute scenes	True	-
TSTAT.S.A001B	`ControlSequenceOfOperation` server attribute	True	-
TSTAT.S.A001C	`SystemMode` server attribute	True	-
TSTAT.S.A001C.Scene	`SystemMode` server attribute scenes	True	-
TSTAT.S.Afffd	`ClusterRevision` server attribute	True	-
TSTAT.S.Afffe	`AttributeReportingStatus` server attribute	False	-
TSTAT.S.C00.Rsp	Setpoint Raise/Lower server command	True	-
TSTAT.S.C01.Rsp	Set Weekly Schedule server command	True	-
TSTAT.S.C02.Rsp	Get Weekly Schedule server command	True	-
TSTAT.S.C03.Rsp	Clear Weekly Schedule server command	True	-
TSTAT.S.C04.Rsp	Get Relay Status Log server command	True	-
TSTAT.S.C00.Tx	Get Weekly Schedule Response server command	True	-
TSTAT.S.C01.Tx	Get Relay Status Log Response server command	True	-
TSTAT.C.Afffd	`ClusterRevision` client attribute	True	-
TSTAT.C.Afffd	`AttributeReportingStatus` client attribute	False	-
TSTAT.C.C00.Rsp	Get Weekly Schedule Response client command	True	-
TSTAT.C.C01.Rsp	Get Relay Status Log Response client command	True	-
TSTAT.C.C00.Tx	Setpoint Raise/Lower client command	True	-
TSTAT.C.C01.Tx	Set Weekly Schedule client command	True	-
TSTAT.C.C02.Tx	Get Weekly Schedule client command	True	-
TSTAT.C.C03.Tx	Clear Weekly Schedule client command	True	-
TSTAT.C.C04.Tx	Get Relay Status Log client command	True	-

5.39.2 Thermostat cluster functions

5.39.2.1 ZbZclThermClientAlloc

```
struct ZbZclClusterT * ZbZclThermClientAlloc(struct ZigBeeT *zb, uint8_t endpoint);
```

Create a new instance of the Scenes Client cluster

Table 640. ZbZclThermClientAlloc parameters

Parameter	Description
zb	Zigbee® stack instance
endpoint	Endpoint on which to create cluster

Return: Cluster pointer, or NULL if there is an error.

5.39.2.2 *ZbZclThermClientClearWeeklySched*

```
enum ZclStatusCodeT ZbZclThermClientClearWeeklySched(struct ZbZclClusterT *cluster, const
struct ZbApsAddrT *dst, void (*callback)(struct ZbZclCommandRspT *rsp, void*arg), void *arg);
```

Send a Clear Weekly Schedule command (optional)

Table 641. ZbZclThermClientClearWeeklySched parameters

Parameter	Description
cluster	Cluster instance from which to send this command
dst	Requested destination address
callback	Callback function that is invoked when the response is received
arg	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.39.2.3 *ZbZclThermClientGetRelayStatusLog*

```
enum ZclStatusCodeT ZbZclThermClientGetRelayStatusLog(struct ZbZclClusterT*cluster, const
struct ZbApsAddrT *dst, void (*callback)(struct ZbZclCommandRspT*rsp, void *arg), void *arg);
```

Send a Get Relay Status Log command (optional)

Table 642. ZbZclThermClientGetRelayStatusLog parameters

Parameter	Description
cluster	Cluster instance from which to send this command
dst	Requested destination address
callback	Callback function that is invoked when the response is received
arg	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.39.2.4 *ZbZclThermClientGetWeeklySched*

```
enum ZclStatusCodeT ZbZclThermClientGetWeeklySched(struct ZbZclClusterT *cluster, const
struct ZbApsAddrT *dst, struct ZbZclThermCliGetWeeklyT *req, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Get Weekly Schedule command (optional)

Table 643. ZbZclThermClientGetWeeklySched parameters

Parameter	Description
cluster	Cluster instance from which to send this command

Parameter	Description
`dst`	Requested destination address
`req`	Thermostat Get Weekly Schedule info
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.39.2.5 *ZbZclThermClientSetWeeklySched*

```
enum ZclStatusCodeT ZbZclThermClientSetWeeklySched(struct ZbZclClusterT *cluster, const
struct ZbApsAddrT *dst, struct ZbZclThermWeeklySchedT *req, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Set Weekly Schedule command (optional)

Table 644. ZbZclThermClientSetWeeklySched parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`req`	Thermostat Weekly Schedule structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.39.2.6 *ZbZclThermClientSetpointRaiseLower*

```
enum ZclStatusCodeT ZbZclThermClientSetpointRaiseLower(struct ZbZclClusterT*cluster, const
struct ZbApsAddrT *dst, struct ZbZclThermCliSetpointT *req, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Setpoint Raise/Lower command

Table 645. ZbZclThermClientSetpointRaiseLower parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`req`	Thermostat Setpoint structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.39.2.7 *ZbZclThermServerAlloc*

```
struct ZbZclClusterT * ZbZclThermServerAlloc(struct ZigBeeT *zb, uint8_t endpoint, struct
ZbZclThermServerCallbacksT *callbacks, void *arg);
```

Create a new instance of the Thermostat Server cluster

Table 646. ZbZclThermServerAlloc parameters

Parameter	Description
zb	Zigbee® stack instance
endpoint	Endpoint on which to create cluster
callbacks	Structure containing any callback function pointers for this cluster
arg	Pointer to application data that is provided back to the callback function when invoked

Return: Cluster pointer, or NULL if there is an error.

5.39.2.8 *ZbZclThermServerGetRelayStatusLogRsp*

```
enum ZclStatusCodeT ZbZclThermServerGetRelayStatusLogRsp(struct ZbZclClusterT*cluster,
struct ZbZclAddrInfoT *dst, struct ZbZclThermSvrGetRelayStatusLogRspT*rsp);
```

Send a Get Relay Status Log Response command

Table 647. ZbZclThermServerGetRelayStatusLogRsp parameters

Parameter	Description
cluster	Cluster instance from which to send this command
dst	Destination address for request
rsp	Get Relay Status Log Response structure

Return: ZCL_STATUS_SUCCESS if successful, or other ZclStatusCodeT value on error.

5.39.2.9 *ZbZclThermServerGetWeeklySchedRsp*

```
enum ZclStatusCodeT ZbZclThermServerGetWeeklySchedRsp(struct ZbZclClusterT*cluster, struct
ZbZclAddrInfoT *dst, struct ZbZclThermWeeklySchedT *rsp);
```

Send a Get Weekly Schedule Response command

Table 648. ZbZclThermServerGetWeeklySchedRsp parameters

Parameter	Description
cluster	Cluster instance from which to send this command
dst	Requested destination address
rsp	Server clear get weekly schedule response

Return: ZCL_STATUS_SUCCESS if successful, or other ZclStatusCodeT value on error.

5.39.3 Thermostat cluster enumerations

5.39.3.1 *ZbZclThermAttrT*

Table 649. Thermostat attribute IDs

Parameter	Description
ZCL_THERM_SVR_ATTR_LOCAL_TEMP	LocalTemperature
ZCL_THERM_SVR_ATTR_OUTDOOR_TEMP	OutdoorTemperature (optional)
ZCL_THERM_SVR_ATTR_OCCUPANCY	Occupancy (optional)

Parameter	Description
ZCL_THERM_SVR_ATTR_ABS_MIN_HEAT_SETPOIN T_LIMIT	AbsMinHeatSetpointLimit (optional)
ZCL_THERM_SVR_ATTR_ABS_MAX_HEAT_SETPOIN T_LIMIT	AbsMaxHeatSetpointLimit (optional)
ZCL_THERM_SVR_ATTR_ABS_MIN_COOL_SETPOIN T_LIMIT	AbsMinCoolSetpointLimit (optional)
ZCL_THERM_SVR_ATTR_ABS_MAX_COOL_SETPOIN T_LIMIT	AbsMaxCoolSetpointLimit (optional)
ZCL_THERM_SVR_ATTR_PI_COOLING_DEMAND	PICoolingDemand (optional)
ZCL_THERM_SVR_ATTR_PI_HEATING_DEMAND	PIHeatingDemand (optional)
ZCL_THERM_SVR_ATTR_HVAC_SYSTYPE_CONFIG	HVACSystemTypeConfiguration (optional)
ZCL_THERM_SVR_ATTR_LOCAL_TEMP_CALIB	LocalTemperatureCalibration (optional)
ZCL_THERM_SVR_ATTR_OCCUP_COOL_SETPOINT	OccupiedCoolingSetpoint
ZCL_THERM_SVR_ATTR_OCCUP_HEAT_SETPOINT	OccupiedHeatingSetpoint
ZCL_THERM_SVR_ATTR_UNOCCUP_COOL_SETPOI NT	UnoccupiedCoolingSetpoint (optional)
ZCL_THERM_SVR_ATTR_UNOCCUP_HEAT_SETPOI NT	UnoccupiedHeatingSetpoint (optional)
ZCL_THERM_SVR_ATTR_MIN_HEAT_SETPOINT	MinHeatSetpointLimit (optional)
ZCL_THERM_SVR_ATTR_MAX_HEAT_SETPOINT	MaxHeatSetpointLimit (optional)
ZCL_THERM_SVR_ATTR_MIN_COOL_SETPOINT	MinCoolSetpointLimit (optional)
ZCL_THERM_SVR_ATTR_MAX_COOL_SETPOINT	MaxCoolSetpointLimit (optional)
ZCL_THERM_SVR_ATTR_MIN_SETPOINT_DEADBAN D	MinSetpointDeadBand (optional)
ZCL_THERM_SVR_ATTR_RMT_SENSE	RemoteSensing (optional)
ZCL_THERM_SVR_ATTR_CONTROL_SEQ_OPER	ControlSequenceOfOperation
ZCL_THERM_SVR_ATTR_SYSTEM_MODE	SystemMode
ZCL_THERM_SVR_ATTR_ALARM_MASK	AlarmMask (optional)
ZCL_THERM_SVR_ATTR_RUNNING_MODE	ThermostatRunningMode (optional)
ZCL_THERM_SVR_ATTR_START_OF_WEEK	StartOfWeek (optional)
ZCL_THERM_SVR_ATTR_NUM_WEEKLY_TRANSITIO NS	NumberOfWeeklyTransitions (optional)
ZCL_THERM_SVR_ATTR_NUM_DAILY_TRANSITIONS	NumberOfDailyTransitions (optional)
ZCL_THERM_SVR_ATTR_TEMP_SETPOINT_HOLD	TemperatureSetpointHold (optional)
ZCL_THERM_SVR_ATTR_TEMP_SETPOINT_HOLD_D UR	TemperatureSetpointHoldDuration (optional)
ZCL_THERM_SVR_ATTR_PROG_OPER_MODE	ThermostatProgrammingOperationMode (optional)
ZCL_THERM_SVR_ATTR_RUNNING_STATE	ThermostatRunningState (optional)
ZCL_THERM_SVR_ATTR_SETPOINT_CHANGE_SRC	SetpointChangeSource
ZCL_THERM_SVR_ATTR_SETPOINT_CHANGE_AMT	SetpointChangeAmount
ZCL_THERM_SVR_ATTR_SETPOINT_CHANGE_TIME STAMP	SetpointChangeSourceTimestamp
ZCL_THERM_SVR_ATTR_OCCUP_SETBACK	OccupiedSetback (optional)
ZCL_THERM_SVR_ATTR_OCCUP_SETBACK_MIN	OccupiedSetbackMin (optional)
ZCL_THERM_SVR_ATTR_OCCUP_SETBACK_MAX	OccupiedSetbackMax (optional)
ZCL_THERM_SVR_ATTR_UNOCCUP_SETBACK	UnoccupiedSetback (optional)

Parameter	Description
ZCL_THERM_SVR_ATTR_UNOCCUP_SETBACK_MIN	`UnoccupiedSetbackMin` (optional)
ZCL_THERM_SVR_ATTR_UNOCCUP_SETBACK_MAX	`UnoccupiedSetbackMax` (optional)
ZCL_THERM_SVR_ATTR_EMERGENCY_HEAT_DELTA	`EmergencyHeatDelta` (optional)
ZCL_THERM_SVR_ATTR_AC_TYPE	`ACType`
ZCL_THERM_SVR_ATTR_AC_CAPACITY	`ACCapacity`
ZCL_THERM_SVR_ATTR_AC_REFIGERANT_TYPE	`ACRefrigerantType`
ZCL_THERM_SVR_ATTR_AC_COMPRESSOR_TYPE	`ACCompressorType`
ZCL_THERM_SVR_ATTR_AC_ERROR_CODE	`ACErrorCode`
ZCL_THERM_SVR_ATTR_AC_LOUVER_POSITION	`ACLouverPosition`
ZCL_THERM_SVR_ATTR_AC_COIL_TEMP	`ACCoilTemperature`
ZCL_THERM_SVR_ATTR_AC_CAPACITY_FORMAT	`ACCapacityFormat`

5.39.4 Thermostat cluster structures

5.39.4.1 *ZbZclThermCliGetWeeklyT*

Thermostat Get Weekly Schedule structure

Table 650. **ZbZclThermCliGetWeeklyT parameters**

Parameter	Description
`uint8_t days_to_return`	Days to Return
`uint8_t mode_to_return`	Mode to Return

5.39.4.2 *ZbZclThermCliSetpointT*

Thermostat Setpoint structure

Table 651. **ZbZclThermCliSetpointT parameters**

Parameter	Description
`uint8_t mode`	Mode
`int8_t amount`	Amount

5.39.4.3 *ZbZclThermServerCallbacksT*

Thermostat Server callbacks configuration

Table 652. **ZbZclThermServerCallbacksT parameters**

Parameter	Description
`setpoint_raise_lower` (callback function pointer)	`enum ZclStatusCodeT (*setpoint_raise_lower)(struct ZbZclClusterT *cluster, void *arg,struct ZbZclThermCliSetpointT *req, struct ZbZclAddrInfoT*srcInfo)` Callback to handle Setpoint Raise/Lower command response
`set_weekly` (callback function pointer)	`Enum ZclStatusCodeT (*set_weekly)(struct ZbZclClusterT *cluster, void *arg,struct ZbZclThermWeeklySchedT *req, struct ZbZclAddrInfoT *srcInfo)` Callback to handle Set Weekly Schedule command response

Parameter	Description
get_weekly (callback function pointer)	Enum ZclStatusCodeT (*get_weekly)(struct ZbZclClusterT *cluster, void *arg,struct ZbZclThermCliGetWeeklyT *req, struct ZbZclAddrInfoT *srcInfo) Callback to handle Get Weekly Schedule command response
clear_weekly (callback function pointer)	enum ZclStatusCodeT (*clear_weekly)(struct ZbZclClusterT *cluster, void *arg,struct ZbZclAddrInfoT *srcInfo) Callback to handle Clear Weekly Schedule command response
get_relay_status_log (callback function pointer)	Enum ZclStatusCodeT (*get_relay_status_log)(struct ZbZclClusterT *cluster, void *arg,struct ZbZclAddrInfoT *srcInfo) Callback to handle Get Relay Status Log command response

5.39.4.4 *ZbZclThermSvrGetRelayStatusLogRspT*

Get Relay Status Log Response structure

Table 653. ZbZclThermSvrGetRelayStatusLogRspT parameters

Parameter	Description
uint16_t time_of_day	Time of dDay
uint16_t relay_status	Relay status
int16_t local_temp	Local temperature
uint8_t humidity_percent	Humidity in percentage
int16_t set_point	Unread entries

5.39.4.5 *ZbZclThermTransitionsT*

Thermostat Transition structure

Table 654. ZbZclThermTransitionsT `parameters

Parameter	Description
uint16_t transition_time	Transition time
int16_t heat_set_point	Heat set point (optional)
int16_t cool_set_point	Cool set point (optional)

5.39.4.6 *ZbZclThermWeeklySchedT*

Thermostat Weekly Schedule structure

Table 655. ZbZclThermWeeklySchedT parameters

Parameter	Description
uint8_t num_transitions	Number of transitions for sequence
uint8_t day_of_week_seq	Day of the week for sequence - for example ZCL_THERM_DAY_OF_WEEK_SUNDAY
uint8_t mode_for_seq	Mode for Sequence - for example ZCL_THERM_MODE_HEAT_SETPOINT_PRESENT
struct ZbZclThermTransitionsT	List of transitions

5.40 Thermostat user interface cluster

```
#include "zcl/general/zcl.therm.ui.h"
```

5.40.1 Thermostat user interface cluster description

Table 656. TThermostat user interface cluster interface PICS code description

PICS Code	Name	Supported	Notes
TSUIC.S	Server	True	-
TSUIC.C	Client	True	-
TSUIC.S.A0000	`TemperatureDisplayMode` server attribute	True	-
TSUIC.S.A0001	`KeypadLockout` server attribute	True	-
TSUIC.S.Afffd	`ClusterRevision` server attribute	True	-
TSUIC.S.Afffe	`AttributeReportingStatus` server attribute	False	-
TSUIC.C.Afffd	`ClusterRevision` client attribute	True	-
TSUIC.C.Afffe	`AttributeReportingStatus` client attribute	False	-

5.40.2 Thermostat user interface cluster functions

5.40.2.1 *ZbZclThermUiClientAlloc*

```
struct ZbZclClusterT * ZbZclThermUiClientAlloc(struct ZigBeeT *zb, uint8_t endpoint);
```

Create a new instance of the Thermostat User Interface Client cluster

Table 657. ZbZclThermUiClientAlloc parameters

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster

Return: Cluster pointer, or NULL if there is an error.

5.40.2.2 *ZbZclThermUiServerAlloc*

```
struct ZbZclClusterT * ZbZclThermUiServerAlloc(struct ZigBeeT *zb, uint8_t endpoint);
```

Create a new instance of the Thermostat User Interface Server cluster

Table 658. ZbZclThermUiServerAlloc parameters

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster

Return: Cluster pointer, or NULL if there is an error.

5.40.3 Thermostat user interface cluster enumerations

5.40.3.1 ZbZclThermUiServerAttrT

Table 659. Thermostat user interface attribute IDs

Parameter	Description
ZCL_THERM_UI_SVR_ATTR_DISPLAY_MODE	TemperatureDisplayMode
ZCL_THERM_UI_SVR_ATTR_KEYPAD_LOCKOUT	KeypadLockout
ZCL_THERM_UI_SVR_ATTR_SCHEDULE_PROG_VIS	ScheduleProgrammingVisibility (optional)

5.41 Time cluster

```
#include "zcl/general/zcl.time.h"
```

5.41.1 Time cluster description

Table 660. Time cluster PICS code description

PICS Code	Name	Supported	Notes
T.S	Server	True	-
T.C	Client	True	-
T.S.A0000	Time server attribute	True	-
T.S.A0001	TimeStatus server attribute	True	-
T.S.A0002	TimeZone server attribute	True	Optional
T.S.A0003	DstStart server attribute	True	Optional
T.S.A0004	DstEnd server attribute	True	Optional
T.S.A0005	DstShift server attribute	True	Optional
T.S.A0006	StandardTime server attribute	True	Optional
T.S.A0007	LocalTime server attribute	True	Optional
T.S.A0008	LastSetTime server attribute	True	Optional
T.S.A0009	ValidUntilTime server attribute	True	Optional
T.S.Afffd	ClusterRevision server attribute	True	-
T.S.Afffe	AttributeReportingStatus server attribute	False	-
T.C.Afffd	ClusterRevision client attribute	True	-
T.C.Afffe	AttributeReportingStatus client attribute	False	-

5.41.2 Time cluster functions

5.41.2.1 ZbZclTimeClientAlloc

```
struct ZbZclClusterT * ZbZclTimeClientAlloc(struct ZigBeeT *zb, uint8_t endpoint);
```

Create a new instance of the Time Client cluster

Table 661. ZbZclTimeClientAlloc parameters

Parameter	Description
zb	Zigbee® stack instance
endpoint	Endpoint on which to create cluster

Return: Cluster pointer, or NULL if there is an error.

5.41.2.2 *ZbZclTimeServerAlloc*

```
struct ZbZclClusterT * ZbZclTimeServerAlloc(struct ZigBeeT *zb, uint8_t endpoint, struct
ZbZclTimeServerCallbacks *callbacks, void *arg);
```

Create a new instance of the Time Server cluster

Table 662. **ZbZclTimeServerAlloc parameters**

Parameter	Description
zb	Zigbee® stack instance
endpoint	Endpoint on which to create cluster
callbacks	Structure containing any callback function pointers for this cluster
arg	Pointer to application data that is provided back to the callback function when invoked

Return: Cluster pointer, or NULL if there is an error.

5.41.2.3 *ZbZclTimeServerCurrentTime*

```
uint32_t ZbZclTimeServerCurrentTime(struct ZbZclClusterT *cluster);
```

Call the get_time callback defined as part of the Time Server callbacks configuration

Table 663. **ZbZclTimeServerCurrentTime parameters**

Parameter	Description
cluster	Cluster instance from which to send this command

Return: Current time

5.41.2.4 *ZbZclTimeServerSetTime*

```
void ZbZclTimeServerSetTime(struct ZbZclClusterT *cluster, uint32_t current_time);
```

Call the set_time callback defined as part of the Time Server callbacks configuration

Table 664. **ZbZclTimeServerSetTime parameters**

Parameter	Description
cluster	Cluster instance from which to send this command
current_time	New current time value to set

Return: Void

5.41.3 Time cluster enumerations

5.41.3.1 *ZbZclTimeSvrAttrT*

Table 665. **Time server attribute IDs**

Parameter	Description
ZCL_TIME_ATTR_TIME	Time

Parameter	Description
ZCL_TIME_ATTR_STATUS	`TimeStatus`
ZCL_TIME_ATTR_TIME_ZONE	`TimeZone` (optional)
ZCL_TIME_ATTR_DST_START	`DstStart` (optional)
ZCL_TIME_ATTR_DST_END	`DstEnd` (optional)
ZCL_TIME_ATTR_DST_SHIFT	`DstShift` (optional)
ZCL_TIME_ATTR_STANDARD_TIME	`StandardTime` (optional)
ZCL_TIME_ATTR_LOCAL_TIME	`LocalTime` (optional)
ZCL_TIME_ATTR_LAST_SET_TIME	`LastSetTime` (optional)
ZCL_TIME_ATTR_VALID_UNTIL_TIME	`ValidUntilTime` (optional)

5.41.4 Time cluster structures

5.41.4.1 *ZbZclTimeServerCallbacks*

Time Server callbacks configuration

Table 666. **ZbZclTimeServerCallbacks parameters**

Parameter	Description
`get_time` (callback function pointer)	`uint32_t (*get_time)(struct ZbZclClusterT *clusterPtr, void *arg)` Callback to application, invoked on receipt of Read Attribute request
`set_time` (callback function pointer)	`void (*set_time)(struct ZbZclClusterT *clusterPtr, uint32_t time_val, void *arg)` Callback to application, invoked on receipt of Write Attribute request for the ZCL_TIME_ATTR_TIME attribute. The `set_time` app callback must also set the ZCL_TIME_ATTR_LAST_SET_TIME attribute if successful.

5.42 Tunneling cluster

```
#include "zcl/se/zcl.tunnel.h"
```

5.42.1 Tunneling cluster description

Table 667. **Tunneling cluster PICS code description**

PICS Code	Name	Supported	Notes
SETUN.S	`Server`	True	-
SETUN.C	`Client`	True	-
SETUN.S.A0000	`CloseTunnelTimeout` server attribute	True	-
SETUN.S.Afffd	`ClusterRevision` server attribute	True	-
SETUN.S.Affe	`AttributeReportingStatus` server attribute	False	-
SETUN.S.C00.Rsp	`RequestTunnel` server command	True	-
SETUN.S.C01.Rsp	`CloseTunnel` server command	True	-
SETUN.S.C02.Rsp	`TransferData` server command	True	-
SETUN.S.C03.Rsp	`TransferDataError` server command	True	-

PICS Code	Name	Supported	Notes
SETUN.S.C00.Tx	RequestTunnelResponse server command	True	-
SETUN.S.C01.Tx	TransferData server command	True	-
SETUN.S.C02.Tx	TransferDataError server command	True	-
SETUN.S.C03.Tx	AckTransferData server command	True	Optional
SETUN.C.Afffd	ClusterRevision client attribute	True	-
SETUN.C.Afffe	AttributeReportingStatus client attribute	False	-
SETUN.C.C00.Rsp	RequestTunnelResponse client command	True	-
SETUN.C.C01.Rsp	TransferData client command	True	-
SETUN.C.C02.Rsp	TransferDataError client command	True	-
SETUN.C.C06.Rsp	TunnelClosureNotification client command	True	Optional
SETUN.C.C00.Tx	RequestTunnel client command	True	-
SETUN.C.C01.Tx	CloseTunnel client command	True	-
SETUN.C.C02.Tx	TransferData client command	True	-
SETUN.C.C03.Tx	TransferDataError client command	True	-
SETUN.C.C04.Tx	AckTransferData client command	True	Optional

5.42.2 Tunneling cluster functions

5.42.2.1 *ZbZclTunnelClientAddProto*

```
enum ZclStatusCodeT ZbZclTunnelClientAddProto(struct ZbZclClusterT *cluster, enum
ZbZclTunnelProtocolT protocol, uint16_t mfr, uint16_t mtu, struct ZbZclTunnelProtoCbT
*callbacks);
```

Add a protocol tunnel to the server

Table 668. **ZbZclTunnelClientAddProto parameters**

Parameter	Description
callback	Callback function that is invoked when the response is received
protocol	Protocol ID enumeration
mfr	Manufacturer ID
mtu	Protocol MTU
callbacks	Protocol callbacks

Return: ZCL_STATUS_SUCCESS if successful, or other ZclStatusCodeT value on error.

5.42.2.2 *ZbZclTunnelClientAlloc*

```
struct ZbZclClusterT * ZbZclTunnelClientAlloc(struct ZigBeeT *zb, uint8_t endpoint);
```

Create a new instance of the Tunneling Client cluster

Table 669. **ZbZclTunnelClientAlloc parameters**

Parameter	Description
zb	Zigbee® stack instance

Parameter	Description
endpoint	Endpoint on which to create cluster

Return: Cluster pointer, or NULL if there is an error.

5.42.2.3 *ZbZclTunnelClientCloseQuietReq*

```
enum ZclStatusCodeT ZbZclTunnelClientCloseQuietReq(struct ZbZclClusterT *cluster);
```

Close the local tunnel without informing the server

Table 670. **ZbZclTunnelClientCloseQuietReq parameters**

Parameter	Description
cluster	Cluster instance from which to send this command

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.42.2.4 *ZbZclTunnelClientCloseReq*

```
enum ZclStatusCodeT ZbZclTunnelClientCloseReq(struct ZbZclClusterT *cluster, void (*callback)
(struct ZbZclCommandRspT *zcl_rsp, void *arg), void *arg);
```

Send a Close Tunnel command

Table 671. **ZbZclTunnelClientCloseReq parameters**

Parameter	Description
cluster	Cluster instance from which to send this command
callback	Callback function that is invoked when the response is received
arg	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.42.2.5 *ZbZclTunnelClientConnectReq*

```
enum ZclStatusCodeT ZbZclTunnelClientConnectReq(struct ZbZclClusterT *cluster, uint64_t
dst_addr, uint8_t dst_ep, void (*callback)(struct ZbZclClusterT *cluster, struct
ZbZclTunnelStateT *state, enum ZbZclTunnelStatusT status, void *arg), void*arg);
```

Send a Request Tunnel command

Table 672. **ZbZclTunnelClientConnectReq parameters**

Parameter	Description
cluster	Cluster instance from which to send this command
dst_addr	Destination address for connection request
dst_endpoint	Destination endpoint for connection request
callback	Callback function to handle response
state	Tunneling Cluster State structure
status	Status of the tunnel
arg	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.42.2.6 *ZbZclTunnelClientSendReq*

```
enum ZclStatusCodeT ZbZclTunnelClientSendReq(struct ZbZclClusterT *cluster, const uint8_t
*data, uint16_t len, void (*callback)(struct ZbZclCommandRspT *rspPr, void*arg), void *arg);
```

Send a Transfer Data command

Table 673. ZbZclTunnelClientSendReq parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`data`	Data to send
`len`	Length of data being sent
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.42.2.7 *ZbZclTunnelServerAddProto*

```
enum ZclStatusCodeT ZbZclTunnelServerAddProto(struct ZbZclClusterT *cluster, enum
ZbZclTunnelProtocolT protocol, uint16_t mfr, uint16_t mtu, struct ZbZclTunnelProtoCbT
*callbacks);
```

Add a protocol tunnel to the server

Table 674. ZbZclTunnelServerAddProto parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`protocol`	Protocol ID enumeration
`mfr`	Manufacturer ID
`mtu`	Protocol MTU
`callback`	Protocol callbacks

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.42.2.8 *ZbZclTunnelServerAlloc*

```
struct ZbZclClusterT * ZbZclTunnelServerAlloc(struct ZigBeeT *zb, uint8_t endpoint);
```

Create a new instance of the Tunneling Server cluster

Table 675. ZbZclTunnelServerAlloc parameters

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster

Return: Cluster pointer, or NULL if there is an error.

5.42.2.9 ZbZclTunnelServerSendAllMatch

```
void ZbZclTunnelServerSendAllMatch(struct ZbZclClusterT *cluster, uint64_t eui, void *data,
unsigned int len);
```

Send data to all tunnel clients with matching EUI

Table 676. ZbZclTunnelServerSendAllMatch parameters

Parameter	Description
cluster	Cluster instance from which to send this command
eui	Extended Unique Identifier (EUI) address
data	Data to send
len	Length of data being sent

Return: Void

5.42.2.10 ZbZclTunnelServerSendto

```
enum ZclStatusCodeT ZbZclTunnelServerSendto(struct ZbZclClusterT *cluster, struct
ZbZclTunnelStateT *state, const uint8_t *data, unsigned int len, enum ZbZclDirectionT
direction, void (*callback)(struct ZbZclCommandRspT *rspPtr, void*arg), void *arg);
```

Send data via the tunnel

Table 677. ZbZclTunnelServerSendto parameters

Parameter	Description
cluster	Cluster instance from which to send this command
state	State of the tunnel
data	Data to send
len	Length of data being sent
direction	Direction to send (ZCL_DIRECTION_TO_SERVER or ZCL_DIRECTION_TO_CLIENT)
callback	Callback function to handle response
arg	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.42.2.11 ZbZclTunnelServerStateFindById

```
struct ZbZclTunnelStateT * ZbZclTunnelServerStateFindById(struct ZbZclClusterT*cluster,
uint16_t tunnel_id);
```

Get tunnel server state by tunnel ID

Table 678. ZbZclTunnelServerStateFindById parameters

Parameter	Description
cluster	Cluster instance from which to send this command
tunnel_id	Tunnel ID

Return: State of the tunnel

5.42.2.12 *ZbZclTunnelStateGetDataLen*

```
uint32_t ZbZclTunnelStateGetDataLen(struct ZbZclTunnelStateT *state, bool clear_data);
```

Get received data length using Tunneling Cluster State structure

Table 679. ZbZclTunnelStateGetDataLen parameters

Parameter	Description
`state`	Tunneling Cluster State structure

Return: Length of data in the buffer

5.42.2.13 *ZbZclTunnelStateGetDataPtr*

```
uint8_t * ZbZclTunnelStateGetDataPtr(struct ZbZclTunnelStateT *state);
```

Get received data using Tunneling Cluster State structure

Table 680. ZbZclTunnelStateGetDataPtr parameters

Parameter	Description
`state`	Tunneling Cluster State structure

Return: Data receive buffer

5.42.2.14 *ZbZclTunnelStateGetId*

```
uint16_t ZbZclTunnelStateGetId(struct ZbZclTunnelStateT *state);
```

Get ID using Tunneling Cluster State structure

Table 681. ZbZclTunnelStateGetId parameters

Parameter	Description
`state`	Tunneling Cluster State structure

Return: Allocated tunnel ID

5.42.2.15 *ZbZclTunnelStateGetProtocol*

```
enum ZbZclTunnelProtocolT ZbZclTunnelStateGetProtocol(struct ZbZclTunnelStateT*state);
```

Get Tunneling Protocol Instance using Tunneling Cluster State structure

Table 682. ZbZclTunnelStateGetProtocol parameters

Parameter	Description
`state`	Tunneling Cluster State structure

Return: Tunneling Protocol Instance structure

5.42.3 Tunneling cluster enumerations

5.42.3.1 *ZbZclTunnelProtocolT*

Table 683. Tunnelling protocol ID enumerations

Parameter	Description
ZCL_TUNNEL_PROTO_DLMS_COSEM	DLMS/COSEM (IEC 62056)
ZCL_TUNNEL_PROTO_IEC_61107	IEC 61107
ZCL_TUNNEL_PROTO_ANSI_C12	ANSI C12
ZCL_TUNNEL_PROTO_M_BUS	M-BUS
ZCL_TUNNEL_PROTO_SML	SML
ZCL_TUNNEL_PROTO_CLIMATE_TALK	`ClimateTalk`
ZCL_TUNNEL_PROTO_GB_HRGP	GB-HRGP
ZCL_TUNNEL_PROTO_IPV4	IP v4
ZCL_TUNNEL_PROTO_IPV6	IP v6
ZCL_TUNNEL_PROTO_MANUFACTURER	Manufacturer-defined protocols
ZCL_TUNNEL_PROTO_RESERVED	Reserved

5.42.3.2 *ZbZclTunnelStatusT*

Table 684. Tunneling status values

Parameter	Description
ZCL_TUNNEL_STATUS_SUCCESS	Success
ZCL_TUNNEL_STATUS_BUSY	Busy
ZCL_TUNNEL_STATUS_NO_RESOURCES	No more tunnel IDs
ZCL_TUNNEL_STATUS_PROTO_UNSUPPORTED	Protocol not supported
ZCL_TUNNEL_STATUS_FLOW_UNSUPPORTED	Flow control not supported

5.42.3.3 *ZbZclTunnelSvrAttrT*

Table 685. Tunneling server attribute IDs

Parameter	Description
ZCL_TUNNEL_ATTR_TIMEOUT	`CloseTunnelTimeout`

5.42.3.4 *ZbZclTunnelXferStatusT*

Table 686. Tunneling transfer data status values

Parameter	Description
ZCL_TUNNEL_XFER_STATUS_NO_TUNNEL	No such tunnel
ZCL_TUNNEL_XFER_STATUS_WRONG_DEVICE	Wrong device
ZCL_TUNNEL_XFER_STATUS_OVERFLOW	Data overflow
ZCL_TUNNEL_XFER_STATUS_NO_RESPONSE	No Response - If zcl_`tunnel`_handle_data returns ZCL_TUNNEL_XFER_STATUS_NO_RESPONSE, don’t send a response

5.42.4 Tunneling cluster structures

5.42.4.1 *ZbZclTunnelProtoCbT*

`ZbZclTunnelServerAddProto` and `ZbZclTunnelClientAddProto` callback data structure

Table 687. **ZbZclTunnelProtoCbT parameters**

Parameter	Description
`request` (callback function pointer)	`enum ZbZclTunnelStatusT (*request)(struct ZbZclClusterT *clusterPtr,struct ZbZclTunnelStateT *statePtr, void *priv)` Callback to the request handler. Only applicable for Tunnel Server cluster.
`input` (callback function pointer)	`void (*input)(struct ZbZclClusterT *clusterPtr, struct ZbZclTunnelStateT *statePtr, void *priv)` Callback to the input handler
`close` (callback function pointer)	`void (*close)(struct ZbZclClusterT *clusterPtr, struct ZbZclTunnelStateT *statePtr, void *priv)` Callback to the close handler (optional). Required if the application needs to keep track of open tunnels, e.g., for persistence
`error` (callback function pointer)	`bool (*error)(struct ZbZclClusterT *clusterPtr, struct ZbZclTunnelStateT *statePtr, void *priv,enum ZbZclTunnelXferStatusT status)` Callback to the error handler (optional). To handle ZCL_TUNNEL_SVR_CMD_ERROR. Return false if tunnel should be closed
`void *priv`	Application private data pointer

5.43 Touchlink cluster

```
#include "zcl/zcl.touchlink.h"
```

5.43.1 Touchlink cluster description

Table 688. **Touchlink PICS code description**

PICS Code	Name	Supported	Notes
TC.S	Server	True	-
TC.C	Client	True	-
TC.S.Afffd	`ClusterRevision` server attribute	True	-
TC.S.C00.Rsp	Scan request server command	True	-
TC.S.C02.Rsp	Device information request server command	True	-
TC.S.C06.Rsp	Identify request server command	True	-
TC.S.C07.Rsp	Reset to factory new request server command	True	-
TC.S.C10.Rsp	Network start request server command	True	-
TC.S.C12.Rsp	Network join router request server command	True	-
TC.S.C14.Rsp	Network join end device request server command	True	-
TC.S.C16.Rsp	Network update request server command	True	-
TC.S.C41.Rsp	Get group identifiers request server command	True	-
TC.S.C42.Rsp	Get endpoint list request server command	True	-
TC.S.C01.Tx	Scan response server command	True	-
TC.S.C03.Tx	Device information response server command	True	-

PICS Code	Name	Supported	Notes
TC.S.C11.Tx	Network start response server command	True	-
TC.S.C13.Tx	Network join router response server command	True	-
TC.S.C15.Tx	Network join end device response server command	True	-
TC.S.C40.Tx	Endpoint information server command	True	-
TC.S.C41.Tx	Get group identifiers response server command	True	-
TC.S.C42.Tx	Get endpoint list response server command	True	-
TC.C.Afffd	`ClusterRevision` client attribute	True	-
TC.C.C01.Rsp	Scan response client command	True	-
TC.C.C03.Rsp	Device information response client command	True	-
TC.C.C11.Rsp	Network start response client command	True	-
TC.C.C13.Rsp	Network join router response client command	True	-
TC.C.C15.Rsp	Network join end device response client command	True	-
TC.C.C40.Rsp	Endpoint information client command	True	-
TC.C.C41.Rsp	Get group identifiers response client command	True	-
TC.C.C42.Rsp	Get endpoint list response client command	True	-
TC.C.C00.Tx	Scan request client command	True	-
TC.C.C02.Tx	Device information request client command	True	-
TC.C.C06.Tx	Identify request client command	True	-
TC.C.C07.Tx	Reset to factory new request client command	True	-
TC.C.C10.Tx	Network start request client command	True	-
TC.C.C12.Tx	Network join router request client command	True	-
TC.C.C14.Tx	Network join end device request client command	True	-
TC.C.C16.Tx	Network update request client command	True	-
TC.C.C41.Tx	Get group identifiers request client command	True	-
TC.C.C42.Tx	Get endpoint list request client command	True	-

5.43.2 Touchlink cluster functions

5.43.2.1 *ZbZclTouchlinkInitiatorGetEpListReq*

```
enum ZclStatusCodeT ZbZclTouchlinkInitiatorGetEpListReq(struct ZigBeeT *zb, struct
ZbTlGetEpListReqCmd *cmd, const struct ZbApsAddrT *dst, void (*callback)(struct
ZbTlGetEpListRspCmd *rsp, void *arg), void *arg);
```

Sends a Touchlink Utility Get Endpoint List Request Command

Table 689. **ZbZclTouchlinkInitiatorGetEpListReq parameters**

Parameter	Description
`zb`	Zigbee® stack instance
`dst`	Requested destination address
`cmd`	Get endpoint list request command structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.43.2.2 *ZbZclTouchlinkInitiatorGetGrpIdReq*

```
enum ZclStatusCodeT ZbZclTouchlinkInitiatorGetGrpIdReq(struct ZigBeeT *zb, struct
ZbTlGetGroupIdsReqCmd *cmd, const struct ZbApsAddrT *dst, void (*callback)(struct
ZbTlGetGroupIdsRspCmd *rsp, void *arg), void *arg);
```

Sends a Touchlink Utility Get Group Identifiers Request Command

Table 690. ZbZclTouchlinkInitiatorGetGrpIdReq parameters

Parameter	Description
`zb`	Zigbee® stack instance
`dst`	Requested destination address
`cmd`	Get Group Identifiers Request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.43.2.3 *ZbZclTouchlinkTargetSendEpInfoCmd*

```
enum ZclStatusCodeT ZbZclTouchlinkTargetSendEpInfoCmd(struct ZigBeeT *zb, uint8_t endpoint,
const struct ZbApsAddrT *dst, void (*callback)(struct ZbZclCommandRspT*zcl_rsp, void *arg),
void *arg);
```

Sends a Touchlink Utility Endpoint Information Command

Table 691. ZbZclTouchlinkTargetSendEpInfoCmd parameters

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint identifier. Must match the endpoint already registered with Touchlink through the `ZbStartup` configuration, otherwise an error is returned.
`dst`	Requested destination address
`callback`	Callback function that is invoked when the response is received

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.43.3 Touchlink cluster structures

5.43.3.1 *ZbTlEndpointList*

Endpoint Information Record Entry

Table 692. ZbTlEndpointList parameters

Parameter	Description
`uint16_t nwkAddr`	Network address
`uint8_t endpoint`	Endpoint identifier
`uint16_t profileId`	Profile identifier
`uint16_t deviceId`	Device identifier

Parameter	Description
`uint8_t version`	Version

5.43.3.2 *ZbTlEpInfoCmd*

Endpoint Information Command

Table 693. ZbTlEpInfoCmd parameters

Parameter	Description
`uint64_t ieeeAddr`	IEEE address
`uint16_t nwkAddr`	Network address
`endpoint`	Endpoint on which to create cluster
`uint16_t profileId`	Profile identifier
`uint16_t deviceId`	Device identifier
`uint8_t version`	Version

5.43.3.3 *ZbTlGetEpListReqCmd*

Get Endpoint List Request command

Table 694. ZbTlGetEpListReqCmd parameters

Parameter	Description
`uint8_t startIdx`	Start index

5.43.3.4 *ZbTlGetEpListRspCmd*

Get Endpoint List Response Command

Table 695. ZbTlGetEpListRspCmd parameters

Parameter	Description
`enum ZclStatusCodeT status`	ZCL Status code
`uint8_t total`	Total
`uint8_t startIdx`	Start index
`uint8_t count`	Count
`struct ZbTlEndpointList endpoint_list`	Endpoint information record entry

5.43.3.5 *ZbTlGetGroupIdsReqCmd*

Get Group Identifiers Request command

Table 696. ZbTlGetGroupIdsReqCmd parameters

Parameter	Description
`uint8_t startIdx`	Start index

5.43.3.6 *ZbTlGetGroupIdsRspCmd*

Get Group Identifiers Response command

Table 697. ZbTlGetGroupIdsRspCmd parameters

Parameter	Description
`enum ZclStatusCodeT status`	ZCL Status code
`uint8_t total`	Total
`uint8_t startIdx`	Start index
`uint8_t count`	Count
`struct ZbTlGroupRecordList record_list`	Group Information Record List

5.43.3.7 *ZbTlGroupRecordList*

Group Information Record list

Table 698. ZbTlGroupRecordList parameters

Parameter	Description
`uint16_t group_list`	Group list
`uint8_t groupType`	Group type - Must be 0

5.43.3.8 *ZbTouchlinkCallbacks*

Zigbee Touchlink callback functions for Touchlink Controller Device Utility commands. These are configured by "struct ZbStartupT" with `ZbStartup()`.

Table 699. ZbTouchlinkCallbacks parameters

Parameter	Description
ep_info_cb (callback function pointer)	`Enum ZclStatusCodeT (*ep_info_cb)(struct ZigBeeT *zb, structZbTlEpInfoCmd *cmd,struct ZbZclAddrInfoT *srcInfo, void *arg)` Endpoint Info callback

5.44 Voice over Zigbee® cluster

```
#include "zcl/general/zcl.voice.h"
```

5.44.1 Voice over Zigbee® cluster description

Table 700. Voice over Zigbee® cluster PICS code description

PICS Code	Name	Supported	Notes
Voice.S	`Server`	True	-
Voice.C	`Client`	True	-
Voice.S.A0000	`CodecType` server attribute	True	-
Voice.S.A0001	`SamplingFrequency` server attribute	True	-
Voice.S.A0002	`SamplingFrequency` server attribute	True	-
Voice.S.A0003	`EstablishmentTimeout` server attribute	True	-
Voice.S.Afffd	`ClusterRevision` server attribute	True	-
Voice.S.Afffe	`AttributeReportingStatus` server attribute	False	-

PICS Code	Name	Supported	Notes
Voice.S.C00.Rsp	Establishment Request server command	True	-
Voice.S.C01.Rsp	Voice Transmission server command	True	-
Voice.S.C02.Rsp	Voice Transmission Completion server command	True	-
Voice.S.C03.Rsp	Control Response server command	True	-
Voice.S.C00.Tx	Establishment Response server command	True	-
Voice.S.C01.Tx	Voice Transmission Response server command	True	-
Voice.S.C02.Tx	Control server command	True	-
Voice.C.Afffd	`ClusterRevision` client attribute	True	-
Voice.C.Affe	`AttributeReportingStatus` client attribute	False	-
Voice.C.C00.Rsp	Establishment Response client command	True	-
Voice.C.C01.Rsp	Voice Transmission Response client command	True	-
Voice.C.C02.Rsp	Control client command	True	-
Voice.C.C00.Tx	Establishment Request client command	True	-
Voice.C.C01.Tx	Voice Transmission client command	True	-
Voice.C.C02.Tx	Voice Transmission Completion client command	True	-
Voice.C.C03.Tx	Control Response client command	True	-

5.44.2 Voice over Zigbee® cluster functions

5.44.2.1 *ZbZclVoiceClientAlloc*

```
struct ZbZclClusterT * ZbZclVoiceClientAlloc(struct ZigBeeT *zb, uint8_t endpoint, struct
zcl_voice_client_callbacks_t *callbacks, void *arg);
```

Create a new instance of the Voice Over Zigbee Client cluster

Table 701. **ZbZclVoiceClientAlloc parameters**

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.44.2.2 *ZbZclVoiceClientEstabReq*

```
enum ZclStatusCodeT ZbZclVoiceClientEstabReq(struct ZbZclClusterT *cluster, const
struct ZbApsAddrT *dst, struct voice_estab_req_t *estab_req, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send an Establishment Request command

Table 702. **ZbZclVoiceClientEstabReq parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address

Parameter	Description
estab_req	Establishment Request command request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.44.2.3 *ZbZclVoiceClientSendControlRsp*

```
enum ZclStatusCodeT ZbZclVoiceClientSendControlRsp(struct ZbZclClusterT *cluster, struct
ZbZclAddrInfoT *dst_info, struct voice_control_rsp_t *rsp);
```

Send an Control command

Table 703. **ZbZclVoiceClientSendControlRsp parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst_info`	Destination address for response, including sequence number and tx options
`rsp`	Control command response structure

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.44.2.4 *ZbZclVoiceServerAlloc*

```
struct ZbZclClusterT * ZbZclVoiceServerAlloc(struct ZigBeeT *zb, uint8_t endpoint, struct
zcl_voice_server_callbacks_t *callbacks, void *arg);
```

Create a new instance of the Voice Over Zigbee Server cluster

Table 704. **ZbZclVoiceServerAlloc parameters**

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster
`callbacks`	Structure containing any callback function pointers for this cluster
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: Cluster pointer, or NULL if there is an error.

5.44.2.5 *ZbZclVoiceServerControlReq*

```
enum ZclStatusCodeT ZbZclVoiceServerControlReq(struct ZbZclClusterT *cluster, const
struct ZbApsAddrT *dst, struct voice_control_t *control_cmd, void (*callback)(struct
ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Control command

Table 705. **ZbZclVoiceServerControlReq prameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address

Parameter	Description
`control_cmd`	Control command request structure
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.44.2.6 *ZbZclVoiceServerSendEstabRsp*

```
enum ZclStatusCodeT ZbZclVoiceServerSendEstabRsp(struct ZbZclClusterT *cluster, struct
ZbZclAddrInfoT *dst_info, struct voice_estab_rsp_t *rsp);
```

Send an Establishment Response command

Table 706. **ZbZclVoiceServerSendEstabRsp parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst_info`	Destination address for response, including sequence number and tx options
`rsp`	Establishment Response command structure

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.44.2.7 *ZbZclVoiceServerSendVoiceTxRsp*

```
enum ZclStatusCodeT ZbZclVoiceServerSendVoiceTxRsp(struct ZbZclClusterT *cluster, struct
ZbZclAddrInfoT *dst, struct voice_voice_tx_rsp_t *rsp);
```

Send a Voice Transmission Response command. The application calls this if it ever encounters an error processing a Voice Transmission packet

Table 707. **ZbZclVoiceServerSendVoiceTxRsp parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst_info`	Destination address for response, including sequence number and TX options
`rsp`	Voice Transmission Response command structure

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.44.2.8 *ZbZclVoiceTxCompletedReq*

```
enum ZclStatusCodeT ZbZclVoiceTxCompletedReq(struct ZbZclClusterT *cluster, const struct
ZbApsAddrT *dst, void (*callback)(struct ZbZclCommandRspT *rsp, void *arg), void *arg);
```

Send a Voice Transmission Complete command

Table 708. **ZbZclVoiceTxCompletedReq parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`callback`	Callback function that is invoked when the response is received

Parameter	Description
arg	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other ZclStatusCodeT value on error.

5.44.2.9 *ZbZclVoiceVoiceTxReq*

```
enum ZclStatusCodeT ZbZclVoiceVoiceTxReq(struct ZbZclClusterT *cluster, const struct
ZbApsAddrT *dst, struct voice_voice_tx_t *voice_tx, void (*callback)(struct ZbZclCommandRspT
*rsp, void *arg), void *arg);
```

Send a Voice Transmission command

Table 709. **ZbZclVoiceVoiceTxReq parameters**

Parameter	Description
Voice Transmission command request structure	Cluster instance from which to send this command
dst	Requested destination address
voice_tx	Voice Transmission command request structure
callback	Callback function that is invoked when the response is received
arg	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other ZclStatusCodeT value on error.

5.44.3 Voice over Zigbee® cluster enumerations

5.44.3.1 *ZbZclVoiceSvrAttrT*

Table 710. **Voice-Over-Zigbee server attribute IDs**

Parameter	Description
ZCL_VOICE_ATTR_CODEC_TYPE	CodecType
ZCL_VOICE_ATTR_SAMP_FREQ	SamplingFrequency
ZCL_VOICE_ATTR_CODECRATE	Codecrate
ZCL_VOICE_ATTR_ESTAB_TIMEOUT	EstablishmentTimeout
ZCL_VOICE_ATTR_CODEC_TYPE_SUB_1	CodecTypeSub1 (optional)
ZCL_VOICE_ATTR_CODEC_TYPE_SUB_2	CodecTypeSub2 (optional)
ZCL_VOICE_ATTR_CODEC_TYPE_SUB_3	CodecTypeSub3 (optional)
ZCL_VOICE_ATTR_COMPRESSION_TYPE	CompressionType (optional)
ZCL_VOICE_ATTR_COMPRESSION_RATE	CompressionRate (optional)
ZCL_VOICE_ATTR_OPTION_FLAGS	OptionFlags (optional)
ZCL_VOICE_ATTR_THRESHOLD	Threshold (optional)

5.44.4 Voice over Zigbee® cluster structures

5.44.4.1 *voice_control_rsp_t*

Control Response command structure

Table 711. voice_control_rsp_t parameters

Parameter	Description
`uint8_t ack_nak`	ACK=0x01 NAK=0x00

5.44.4.2 *voice_control_t*

Control command structure

Table 712. voice_control_t parameters

Parameter	Description
`uint8_t control_type`	Control type

5.44.4.3 *voice_estab_req_t*

Establishment Request command structure

Table 713. voice_estab_req_t parameters

Parameter	Description
`uint8_t flag`	Flag
`uint8_t codec_type`	Codec type
`uint8_t samp_freq`	Sample freq.
`uint8_t codec_rate`	Codec rate
`uint8_t service_type`	Service type
`uint8_t codec_type_s1`	Codec type S1
`uint8_t codec_type_s2`	Codec type S2
`uint8_t codec_type_s3`	Codec type S3
`uint8_t comp_type`	Comp. type
`uint8_t comp_rate`	Comp. rate

5.44.4.4 *voice_estab_rsp_t*

Establishment Response command structure

Table 714. voice_estab_rsp_t parameters

Parameter	Description
`uint8_t ack_nak`	ACK=0x01 NAK=0x00
`uint8_t codec_type`	CodecType

5.44.4.5 *voice_voice_tx_rsp_t*

Voice Transmission Response command structure

Table 715. voice_voice_tx_rsp_t parameters

Parameter	Description
`uint8_t error_flag`	Error Flag - for example ZCL_VOICE_TX_RSP_ERROR_DECODE

5.44.4.6 *voice_voice_tx_t*

Voice Transmission command structure

Table 716. **voice_voice_tx_t parameters**

Parameter	Description
`uint8_t *voice_data`	Voice Data
`uint16_t voice_data_len`	Voice Data Length

5.44.4.7 *zcl_voice_client_callbacks_t*

Voice Over Zigbee Client callbacks configuration

Table 717. **zcl_voice_client_callbacks_t parameters**

Parameter	Description
`control` (callback function pointer)	`Enum ZclStatusCodeT (*control)(struct ZbZclClusterT*clusterPtr, struct voice_control_t *cmd_req,struct ZbZclAddrInfoT *src_info, void *arg)` Callback to application, invoked on receipt of Control command

5.44.4.8 *zcl_voice_server_callbacks_t*

Voice Over Zigbee Server callbacks configuration

Table 718. **zcl_voice_server_callbacks_t parameters**

Parameter	Description
`estab_req` (callback function pointer)	`Enum ZclStatusCodeT (*estab_req)(struct ZbZclClusterT *clusterPtr, struct voice_estab_req_t *cmd_req,struct ZbZclAddrInfoT *src_info, void *arg)` Callback to application, invoked on receipt of Establishment Request command
`voice_tx` (callback function pointer)	`enum ZclStatusCodeT (*voice_tx)(struct ZbZclClusterT *clusterPtr, struct voice_voice_tx_t *cmd_req,struct ZbZclAddrInfoT *src_info, void *arg)` Callback to application, invoked on receipt of Voice Transmission command. The application should return ZCL_STATUS_SUCCESS or call `ZbZclVoiceServerSendVoiceTxRsp` to send an error response and return ZCL_STATUS_SUCCESS_NO_DEFAULT_RESPONSE
`tx_complete` (callback function pointer)	`enum ZclStatusCodeT (*tx_complete)(struct ZbZclClusterT *clusterPtr,struct ZbZclAddrInfoT *src_info, void *arg)` Callback to application, invoked on receipt of Voice Transmission Completion command

5.45 Water content measurement cluster

```
#include "zcl/general/zcl.wcm.h"
```

5.45.1 Water content measurement cluster description

Table 719. **Water content measurement cluster PICS code description**

PICS Code	Name	Supported	Notes
WCM.S	Server	True	-
WCM.C	Client	True	-

PICS Code	Name	Supported	Notes
WCM.S.A0000	`MeasuredValue` server attribute	True	-
WCM.S.A0000.Report	`MeasuredValue` server attribute reports	True	-
WCM.S.A0001	`MinMeasuredValue` server attribute	True	-
WCM.S.A0002	`MaxMeasuredValue` server attribute	True	-
WCM.S.Afffd	`ClusterRevision` server attribute	True	-
WCM.S.Afffe	`AttributeReportingStatus` server attribute	False	-
WCM.C.A0000.Report.Rsp	`MeasuredValue` client attribute reports	False	-
WCM.C.Afffd	`ClusterRevision` client attribute	True	-
WCM.C.Afffe	`AttributeReportingStatus` client attribute	False	-

5.45.2 Water content measurement cluster functions

5.45.2.1 *ZbZclWaterContentMeasClientAlloc*

```
struct ZbZclClusterT * ZbZclWaterContentMeasClientAlloc(struct ZigBeeT *zb, uint8_t
endpoint, enum ZbZclClusterIdT clusterID);
```

Create a new instance of the Water Content Measurement Client cluster

Table 720. **ZbZclWaterContentMeasClientAlloc parameters**

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster
`cluster_id`	Cluster identifier (Relative Humidity 0x0405, Leaf Wetness 0x0407 or Soil Moisture 0x0408)

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.45.2.2 *ZbZclWaterContentMeasServerAlloc*

```
struct ZbZclClusterT * ZbZclWaterContentMeasServerAlloc(struct ZigBeeT *zb, uint8_t
endpoint, enum ZbZclClusterIdT clusterID, uint16_t min, uint16_t max);
```

Create a new instance of the Water Content Measurement Server cluster

Table 721. **ZbZclWaterContentMeasServerAlloc parameters**

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster
`clusterID`	Cluster identifier (Relative Humidity 0x0405, Leaf Wetness 0x0407 or Soil Moisture 0x0408)
`min`	Default minimum value
`max`	Default maximum value

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.45.3 Water content measurement cluster enumerations

5.45.3.1 *ZbZclWcmSvrAttrT*

Table 722. **Water Content Measurement server attribute IDs**

Parameter	Description
ZCL_WC_MEAS_ATTR_MEAS_VAL	Measured value
ZCL_WC_MEAS_ATTR_MIN_MEAS_VAL	Minimum measured value
ZCL_WC_MEAS_ATTR_MAX_MEAS_VAL	Maximum measured value
ZCL_WC_MEAS_ATTR_TOLERANCE	Tolerance (optional)

5.46 Window covering cluster

```
#include "zcl/general/zcl.window.h"
```

5.46.1 Window covering cluster description

Table 723. **Window covering PICS code description**

PICS Code	Name	Supported	Notes
WNCV.S	Server	True	-
WNCV.C	Client	True	-
WNCV.S.Lift	Lift Action Support	True	-
WNCV.S.Tilt	Tilt Action Support	True	-
WNCV.S.ClosedLoop	Closed Loop Control server command	False	-
WNCV.S.A0000	`WindowCoveringType` server attribute	True	-
WNCV.S.A0007	`ConfigStatus` server attribute	True	-
WNCV.S.A0008	`CurrentPositionLiftPercentage` server attribute	True	Mandatory only if Closed Loop control is enabled
WNCV.S.A0008.Scene	`CurrentPositionLiftPercentage` server attribute scenes	False	-
WNCV.S.A0008.Report.Default Config	`CurrentPositionLiftPercentage` server attribute reports	False	-
WNCV.S.A0008.Report.Tx	`CurrentPositionLiftPercentage` server attribute reports	False	-
WNCV.S.A0009	`CurrentPositionTiltPercentage` server attribute	True	Mandatory only if Closed Loop control is enabled
WNCV.S.A0009.Scene	`CurrentPositionTiltPercentage` server attribute scenes	False	-
WNCV.S.A0009.Report.Default Config	`CurrentPositionTiltPercentage` server attribute reports	False	-
WNCV.S.A0009.Report.Tx	`CurrentPositionTiltPercentage` server attribute reports	False	-
WNCV.S.A0010	`InstalledOpenLimitLift` server attribute	False	Mandatory only if Closed Loop control is enabled

PICS Code	Name	Supported	Notes
WNCV.S.A0011	InstalledClosedLimitLift server attribute	False	Mandatory only if Closed Loop control is enabled
WNCV.S.A0012	InstalledOpenLimitTilt server attribute	False	Mandatory only if Closed Loop control is enabled
WNCV.S.A0013	InstalledClosedLimitTilt server attribute	False	Mandatory only if Closed Loop control is enabled
WNCV.S.A0017	Mode server attribute	True	-
WNCV.S.Afffd	ClusterRevision server attribute	True	-
WNCV.S.Afffe	AttributeReportingStatus server attribute	False	-
WNCV.S.C00.Rsp	Up / Open server command	True	-
WNCV.S.C01.Rsp	Down / Close server command	True	-
WNCV.S.C02.Rsp	Stop server command	True	-
WNCV.S.C04.Rsp	Go To Lift Value server command	False	-
WNCV.S.C05.Rsp	Go to Lift Percentage server command	False	-
WNCV.S.C07.Rsp	Go to Tilt Value server command	False	-
WNCV.S.C08.Rsp	Go to Tilt Percentage server command	False	-
WNCV.C.A0008.Report.Rsp	CurrentPositionLiftPercentage client attribute reports	False	-
WNCV.C.A0009.Report.Rsp	CurrentPositionTiltPercentage client attribute reports	False	-
WNCV.C.Afffd	ClusterRevision client attribute	True	-
WNCV.C.Afffe	AttributeReportingStatus client attribute	False	-
WNCV.C.C00.Tx	Up / Open client command	True	-
WNCV.C.C01.Tx	Down / Close client command	True	-
WNCV.C.C02.Tx	Stop client command	True	-
WNCV.C.C04.Tx	Go To Lift Value client command	False	-
WNCV.C.C05.Tx	Go to Lift Percentage client command	False	-
WNCV.C.C07.Tx	Go to Tilt Value client command	False	-
WNCV.C.C08.Tx	Go to Tilt Percentage client command	False	-

5.46.2 Window covering cluster functions

5.46.2.1 *ZbZclWindowClientAlloc*

```
struct ZbZclClusterT * ZbZclWindowClientAlloc(struct ZigBeeT *zb, uint8_t endpoint);
```

Create a new instance of the Window Covering Client cluster

Table 724. **ZbZclWindowClientAlloc parameters**

Parameter	Description
zb	Zigbee® stack instance

Parameter	Description
`endpoint`	Endpoint on which to create cluster

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.46.2.2 *ZbZclWindowClientCommandDown*

```
enum ZclStatusCodeT ZbZclWindowClientCommandDown(struct ZbZclClusterT *cluster, const struct
ZbApsAddrT *dst, void (*callback)(struct ZbZclCommandRspT *zcl_rsp, void *arg), void *arg);
```

Send a Down/Close command

Table 725. **ZbZclWindowClientCommandDown parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.46.2.3 *ZbZclWindowClientCommandStop*

```
enum ZclStatusCodeT ZbZclWindowClientCommandStop(struct ZbZclClusterT *cluster, const struct
ZbApsAddrT *dst, void (*callback)(struct ZbZclCommandRspT *zcl_rsp, void *arg), void *arg);
```

Send a Stop command

Table 726. **ZbZclWindowClientCommandStop parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.46.2.4 *ZbZclWindowClientCommandUp*

```
enum ZclStatusCodeT ZbZclWindowClientCommandUp(struct ZbZclClusterT *cluster, const struct
ZbApsAddrT *dst, void (*callback)(struct ZbZclCommandRspT *zcl_rsp, void*arg), void *arg);
```

Send an Up/Open command

Table 727. **ZbZclWindowClientCommandUp parameters**

Parameter	Description
`cluster`	Cluster instance from which to send this command
`dst`	Requested destination address
`callback`	Callback function that is invoked when the response is received
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.46.2.5 *ZbZclWindowClosureServerMode*

```
enum ZclStatusCodeT ZbZclWindowClosureServerMode(struct ZbZclClusterT *cluster, uint8_t
mode);
```

Configure the Window Covering mode

Table 728. ZbZclWindowClosureServerMode parameters

Parameter	Description
`cluster`	Cluster instance from which to send this command
`mode`	Window Covering mode bit mask

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.46.2.6 *ZbZclWindowServerAlloc*

```
struct ZbZclClusterT * ZbZclWindowServerAlloc(struct ZigBeeT *zb, uint8_t endpoint, struct
ZbZclWindowServerCallbacksT *callbacks, void *arg);
```

Create a new instance of the Window Covering Server cluster

Table 729. ZbZclWindowServerAlloc parameters

Parameter	Description
`zb`	Zigbee® stack instance
`endpoint`	Endpoint on which to create cluster
`callbacks`	Structure containing any callback function pointers for this cluster
`arg`	Pointer to application data that is provided back to the callback function when invoked

Return: ZCL_STATUS_SUCCESS if successful, or other `ZclStatusCodeT` value on error.

5.46.3 Window covering cluster enumerations

5.46.3.1 *ZbZclWncvServerAttrT*

Table 730. Window covering server attribute IDs

Covering server attributes	Description
ZCL_WNCV_SVR_ATTR_COVERING_TYPE	`WindowCoveringType`
ZCL_WNCV_SVR_ATTR_PHY_CLOSE_LIMIT_LIFT	`PhysicalClosedLimitLift` (optional)
ZCL_WNCV_SVR_ATTR_PHY_CLOSE_LIMIT_TILT	`PhysicalClosedLimitTilt` (optional)
ZCL_WNCV_SVR_ATTR_CURR_POSITION_LIFT	`CurrentPositionLift` (optional)
ZCL_WNCV_SVR_ATTR_CURR_POSITION_TILT	`CurrentPositionTilt` (optional)
ZCL_WNCV_SVR_ATTR_ACTUATION_NUMBER_LIFT	`NumberOfActuationsLift` (optional)
ZCL_WNCV_SVR_ATTR_ACCUATION_NUMBER_TILT	`NumberOfActuationsTilt` (optional)
ZCL_WNCV_SVR_ATTR_CONFIG_STATUS	`ConfigStatus`
ZCL_WNCV_SVR_ATTR_CURR_POS_LIFT_PERCENT	`CurrentPositionLiftPercentage`
ZCL_WNCV_SVR_ATTR_CURR_POS_TILT_PERCENT	`CurrentPositionTiltPercentage`

Covering server attributes	Description
ZCL_WNCV_SVR_ATTR_INSTALLED_OPENED_LIMIT_LIFT	InstalledOpenLimitLift
ZCL_WNCV_SVR_ATTR_INSTALLED_CLOSED_LIMIT_LIFT	InstalledClosedLimitLift
ZCL_WNCV_SVR_ATTR_INSTALLED_OPENED_LIMIT_TILT	InstalledOpenLimitTilt
ZCL_WNCV_SVR_ATTR_INSTALLED_CLOSED_LIMIT_TILT	InstalledClosedLimitTilt
ZCL_WNCV_SVR_ATTR_VELOCITY_LIFT	VelocityLift (optional)
ZCL_WNCV_SVR_ATTR_ACCELERATION_TIME_LIFT	AccelerationTimeLift (optional)
ZCL_WNCV_SVR_ATTR_DECELERATION_TIME_LIFT	DecelerationTimeLift (optional)
ZCL_WNCV_SVR_ATTR_MODE	Mode
ZCL_WNCV_SVR_ATTR_INTERMEDIATE_SETPOINT S_LIFT	Intermediate Setpoints - Lift (optional)
ZCL_WNCV_SVR_ATTR_INTERMEDIATE_SETPOINT S_TILT	Intermediate Setpoints - Tilt (optional)

5.46.3.2 *ZbZclWncvTypes*

Table 731. **Window covering type enumerations**

Enumeration types	Description
ZCL_WNCV_TYPE_ROLLERSHADE	Rollershade
ZCL_WNCV_TYPE_ROLLERSHADE_2_MOTOR	Rollershade - 2 Motor
ZCL_WNCV_TYPE_ROLLERSHADE_EXTERIOR	Rollershade – Exterior
ZCL_WNCV_TYPE_ROLLERSHADE_EXTERIOR_2_M OTOR	Rollershade - Exterior - 2 Motor
ZCL_WNCV_TYPE_DRAPERY	Drapery
ZCL_WNCV_TYPE_AWNING	Awning
ZCL_WNCV_TYPE_SHUTTER	Shutter
ZCL_WNCV_TYPE_TILT_BLIND_TILT_ONLY	Tilt Blind - Tilt Only
ZCL_WNCV_TYPE_TILT_BLIND_LIFT_TILT	Tilt Blind - Lift and Tilt
ZCL_WNCV_TYPE_PROJECTOR_SCREEN	Projector Screen

5.46.4 Window covering cluster structures

5.46.4.1 *ZbZclWindowServerCallbacksT*

Window Covering Server callbacks configuration

Table 732. **ZbZclWindowServerCallbacksT parameters**

Parameter	Description
up_command (callback function pointer)	`Enum ZclStatusCodeT (*up_command)(struct ZbZclClusterT *cluster,struct ZbZclHeaderT *zclHdrPtr, struct ZbApsdeDataIndT *dataIndPtr, void *arg)` Callback to application, invoked on receipt of Up/Open command. Application should update ZCL_WNCV_SVR_ATTR_CURR_POS_LIFT_PERCENT and ZCL_WNCV_SVR_ATTR_CURR_POS_TILT_PERCENT.

Parameter	Description
`down_command` (callback function pointer)	`enum ZclStatusCodeT (*down_command)(struct ZbZclClusterT *cluster,struct ZbZclHeaderT *zclHdrPtr, struct ZbApsdeDataIndT *dataIndPtr, void *arg)` Callback to application, invoked on receipt of Down/Close command. Application should update ZCL_WNCV_SVR_ATTR_CURR_POS_LIFT_PERCENT and ZCL_WNCV_SVR_ATTR_CURR_POS_TILT_PERCENT.
`stop_command` (callback function pointer)	`enum ZclStatusCodeT (*stop_command)(struct ZbZclClusterT *cluster,struct ZbZclHeaderT *zclHdrPtr, struct ZbApsdeDataIndT *dataIndPtr, void *arg)` Callback to application, invoked on receipt of Stop command. Application should update ZCL_WNCV_SVR_ATTR_CURR_POS_LIFT_PERCENT and ZCL_WNCV_SVR_ATTR_CURR_POS_TILT_PERCENT.
`set_lift_and_tilt_command` (callback function pointer)	`enum ZclStatusCodeT (*set_lift_and_tilt_command)(struct ZbZclClusterT *cluster,void *arg, uint8_t liftPercentage, uint8_t tiltPercentage)` Callback to application, invoked to handle setting a Scene, which includes Lift and Tilt values. Application should update ZCL_WNCV_SVR_ATTR_CURR_POS_LIFT_PERCENT and ZCL_WNCV_SVR_ATTR_CURR_POS_TILT_PERCENT.

Revision history

Table 733. Document revision history

Date	Version	Changes
17-Feb-2022	1	Initial release.

Contents

List of tables