Cluster ID	Name
0x8041	Security_Retrieve_Authentication_Token_rsp
0x8043	Security_Set_Configuration_rsp

Devices supporting the commands in Table 2-42 SHALL be able to handle fragmentation and reassembly of these commands. The required parameters for fragmentation are specified in section 2.2.8.4.5.2. Additionally, the Fragmentation Parameters Global TLV can be used to advertise a device's capabilities. This TLV is mandatory to be included in various messages as described in the description of those ZDO messages.

A sending device SHALL determine the receiving device's fragmentation capabilities prior to sending it a fragmentation transmission. For devices already on the network this can be done by querying the Node Descriptor using the Node_Desc_req. For devices not on the network yet, the Trust Center includes the Fragmentation Parameters Global TLV in the set of TLVs advertised in the Beacons of the Network. This is updated in all routers via the Mgmt_Permit_Joining_req.

A device sending a ZDO Response SHALL assume the device that sent the request can support fragmentation. The device sending the response determines the fragmented transmission size based on its capabilities, the requestor's capabilities, and the default minimum.

The Responder SHALL determine the maximum incoming transfer size of the Requester in the following way.

1. If the Requester provided a Fragmentation Parameters Global TLV in the request, the Maximum Incoming Transfer Size from the TLV SHALL be used. If it is not provided in the request, the default Maximum Incoming transfer size of 128 bytes SHALL be used.
2. Compare the value determined in step 1 to the device's local Maximum Outgoing Transfer Size. Take the smaller of the two values.

If the response is larger than the requesting device can handle, then a ZDO response with a status of FRAME_TOO_LARGE is generated.

For example, Device A sends a ZDO_Security_Get_Configuration_req and indicates via the Fragmentation Parameters Global TLV it supports up to 200 bytes for its Maximum Incoming Transfer Size. Device B prepares a ZDO Security_Get_Configuration_rsp and examines its own local Maximum Outgoing Transfer Size, which is 300. It uses the smaller value of 200 indicated by Device A when fragmenting the transmission. If the response would exceed Device A's smaller value it would instead generate a ZDO Security_Get_Configuration_rsp with a status of FRAME_TOO_LARGE.

2.4.2.8.4 APS Acknowledgements

All unicast ZDO Command request and responses SHALL set the Acknowledgement request sub-field of the APS Frame control. This will enable ZDO messages to overcome transient routing or buffering failures in the network. This SHALL be done by submitting a APSDE-DATA.request with the TxOptions including 0x04 in the value. When the ZDO message allows fragmentation the options SHALL also include 0x08, fragmentation permitted.

2.4.3 Client Services

The Device Profile Client Services support the transport of device and service discovery requests, bind requests, unbind requests, and network management requests from client to server. Additionally, Client Services support receipt of responses to these requests from the server.

Restricted Mode (apsZdoRestrictedMode) is a mode where a device will conditionally accept specific ZDO commands, depending on the restricted criteria, source address, and encryption policy of the incoming command. If a command is accepted, it is subject to normal command processing. The acceptance criteria is explain further below:

1. If the command is marked as "Yes" in the *Restricted Command* column, do the following:
   a. If *apsZdoRestrictedMode* in the AIB is set to FALSE, the command is not restricted.
      i. Go to Step 2.
   b. If the sender is the Trust Center AND has APS encryption, the command is not restricted.

i. Go to Step 2.

c. Otherwise, the command SHALL NOT be processed. The receiver SHALL do the following:

i. If the command was broadcast, no error is generated.

1. No more processing is done.

ii. If the command was unicast, generate an error message. Create the corresponding ZDO Response frame with a status of NOT_AUTHORIZED.

1. No more processing is done.

2. Continue processing the command normally.

2.4.3.1 Device and Service Discovery Client Services

Table 2-43 lists the commands supported by Device Profile, Device, and Service Discovery Client Services. Each of these commands will be discussed in the following sections.

Table 2-43. Device and Service Discovery Client Services Commands

Device and Service Discovery Client Services	Cluster ID	Client Transmission	Server Processing	Restricted Command
NWK_addr_req	0x0000	O	M	No
IEEE_addr_req	0x0001	O	M	No
Node_Desc_req	0x0002	M	M	No
Power_Desc_req	0x0003	O	M	No
Simple_Desc_req	0x0004	O	M	No
Active_EP_req	0x0005	O	M	No
Match_Desc_req	0x0006	O	M	No
Complex_Desc_req	0x0010	Deprecated	Deprecated	-
User_Desc_req	0x0011	Deprecated	Deprecated	-
Discovery_Cache_req	0x0012	Deprecated	Deprecated	-
Device_annce	0x0013	O	M	No
Parent_annce	0x001F	M	M	No
User_Desc_set	0x0014	Deprecated	Deprecated	-
System_Server_Discovery_req	0x0015	O	M	No
Discovery_store_req	0x0016	Deprecated	Deprecated	-
Node_Desc_store_req	0x0017	Deprecated	Deprecated	-

Device and Service Discovery Client Services	Cluster ID	Client Transmission	Server Processing	Restricted Command
Power_Desc_store_req	0x0018	Deprecated	Deprecated	-
Active_EP_store_req	0x0019	Deprecated	Deprecated	-
Simple_Desc_store_req	0x001a	Deprecated	Deprecated	-
Remove_node_cache_req	0x001b	Deprecated	Deprecated	-
Find_node_cache_req	0x001c	Deprecated	Deprecated	-
Extended_Simple_Desc_req	0x001d	Deprecated	Deprecated	-
Extended_Active_EP_req	0x001e	Deprecated	Deprecated	-

2.4.3.1.1 NWK_addr_req

The NWK_addr_req command (ClusterID=0x0000) SHALL be formatted as illustrated in Figure 2-18.

Octets: 8	1	1
IEEEAddress	RequestType	StartIndex

Figure 2-18. Format of the NWK_addr_req Command Frame

Table 2-44 specifies the fields of the NWK_addr_req Command Frame.

Table 2-44. Fields of the NWK_addr_req Command Frame

Name	Type	Valid Range	Description
IEEEAddr	IEEE Address	A valid 64-bit IEEE address	The IEEE address to be matched by the Remote Device
RequestType	Integer	0x00 – 0xff	Request type for this command: 0x00 – Single device response 0x01 – Extended response 0x02-0xFF – reserved
StartIndex	Integer	0x00 – 0xff	If the Request type for this command is Extended response, the StartIndex provides the starting index for the requested elements of the associated devices list

2.4.3.1.1.1 When Generated

The NWK_addr_req is generated from a Local Device wishing to inquire as to the 16-bit address of the Remote Device based on its known IEEE address. The destination addressing on this command SHALL be unicast or broadcast to all devices for which macRxOnWhenIdle = TRUE.

2.4.3.1.1.2 Effect on Receipt

Upon receipt, a Remote Device SHALL compare the IEEEAddr to its *nwkIeeeAddress* in the NIB or any IEEE address held in its *nwkNeighborTable* where the Device Type field of the entry is 0x02 (End Device). If there is no match and the request was unicast, a NWK_addr_rsp command SHALL be generated and sent back to the local device with the Status field set to DEVICE_NOT_FOUND, the IEEEAddrRemoteDev field set to the IEEE address of the request; the NWKAddrRemoteDev field set to 0xFFFF indicating that there is no known short address; and the NumAssocDev, StartIndex, and NWKAddrAssocDevList fields SHALL NOT be included in the frame. If there is no match and the command was received as a broadcast, the request SHALL be discarded and no response generated. Note that router parent *and* the macRxOnWhenIdle=TRUE end device SHALL *both* respond to the NWK Address request when the request is sent to the macRxOnWhenIdle=TRUE broadcast address.

If a match is detected between the contained IEEEAddr and the receiving device's *nwkIeeeAddress* or one held in the receiving device's *nwkNeighborTable*, the RequestType SHALL be used to create a response. If the RequestType is one of the reserved values and the request was not sent to a broadcast address, a NWK_addr_rsp command SHALL be generated and sent back to the local device with the Status field set to INV_REQUESTTYPE; the IEEEAddrRemoteDev field set to the IEEE address of the request; the NWKAddrRemoteDev field set to the network address corresponding to the IEEE address in the request; the NumAssocDev, StartIndex, and NWKAddrAssocDevList fields SHALL NOT be included in the frame.

If the RequestType is single device response, a NWK_addr_rsp command SHALL be generated and sent back to the local device with the Status field set to SUCCESS, the IEEEAddrRemoteDev field set to the IEEE address of the request; the NWKAddrRemoteDev field set to the NWK address of the discovered device; and the NumAssocDev, StartIndex, and NWKAddrAssocDevList fields SHALL NOT be included in the frame.

If the RequestType was Extended response and the Remote Device is either the Zigbee coordinator or router, a NWK_addr_rsp command SHALL be generated and sent back to the local device with the Status field set to SUCCESS, the IEEEAddrRemoteDev field set to the IEEE address of the device itself, and the NWKAddrRemoteDev field set to the NWK address of the device itself. The Remote Device SHALL also supply a list of all 16-bit NWK addresses in the NWKAddrAssocDevList field, starting with the entry StartIndex and continuing with whole entries until the maximum APS packet length is reached, for all devices in its *nwkNeighborTable* where the Device Type is 0x02 (End Device). It SHALL then set the NumAssocDev field to the number of entries in the NWKAddrAssocDevList field.

2.4.3.1.2 IEEE_addr_req

The IEEE_addr_req command (ClusterID=0x0001) SHALL be formatted as illustrated in Figure 2-19.

Octets: 2	**1**	**1**
NWKAddrOfInterest	RequestType	StartIndex

Figure 2-19. Format of the IEEE_addr_req_Command Frame

Table 2-45 specifies the fields of the IEEE_addr_req command frame.

Table 2-45. Fields of the IEEE_addr_req Command

Name	Type	Valid Range	Description
NWKAddrOfInterest	Device Address	16-bit NWK address	NWK address that is used for IEEE address mapping.
RequestType	Integer	0x00-0xff	Request type for this command: 0x00 – Single device response 0x01 – Extended response 0x02-0xff – reserved
StartIndex	Integer	0x00-0xff	If the Request type for this command is Extended response, the StartIndex provides the starting index for the requested elements of the associated devices list.

2.4.3.1.2.1 When Generated

The IEEE_addr_req is generated from a Local Device wishing to inquire as to the 64-bit IEEE address of the Remote Device based on their known 16-bit address. The destination addressing on this command SHALL be unicast. or broadcast to all devices for which macRxOnWhenIdle = TRUE.

2.4.3.1.2.2 Effect on Receipt

Upon receipt a Remote Device SHALL compare the NWKAddrOfInterest to its local *nwkNetworkAddress* value in the NIB, or compare any Network address field held in its *nwkNeighborTable* that also has the Device Type field set to 0x02 (End Device). If there is no match, an IEEE_addr_rsp command SHALL be generated and sent back to the local device with the Status field set to DEVICE_NOT_FOUND; theIEEEAddrRemoteDev field set to the IEEE address of 0xFFFFFFFFFFFFFFFF; the NWKAddrRemoteDev field set to the NWK address of the request; and the NumAssocDev, StartIndex, and NWKAddrAssocDevList fields SHALL NOT be included in the frame.

If a match is detected between the contained NWKAddrOfInterest and the receiving device's *nwkNetworkAddress* or one held in the *nwkNeighborTable*, the RequestType SHALL be used to create a response. If the RequestType is one of the reserved values, an IEEE_addr_rsp command SHALL be generated and sent back to the local device with the Status field set to INV_REQUESTTYPE, the IEEEAddrRemoteDev field set to the IEEE address of this device, the NWKAddrRemoteDev field set to the network address of this device and the NumAssocDev, StartIndex, and NWKAddrAssocDevList fields SHALL NOT be included in the frame.

If the RequestType is single device response, an IEEE_addr_rsp command SHALL be generated and sent back to the local device with the Status field set to SUCCESS, the IEEEAddrRemoteDev field set to the IEEE address of the discovered device, the NWKAddrRemoteDev field set to the NWK address of the request and the NumAssocDev, StartIndex, and NWKAddrAssocDevList fields SHALL NOT be included in the frame.

If the RequestType indicates an Extended Response and the Remote Device is the Zigbee coordinator or router with associated devices, an IEEE_addr_rsp command SHALL be generated and sent back to the local device with the Status field set to SUCCESS, the IEEEAddrRemoteDev field set to the IEEE address of the device itself, and the NWKAddrRemoteDev field set to the NWK address of the device itself. The Remote Device SHALL also supply a list of all 16-bit network addresses in the NWKAddrAssocDevList field, starting with the entry StartIndex and continuing with whole entries until the maximum APS packet length is reached, for each entry in the *nwkNeighborTable* where the Device Type field is set to 0x02 (End Device). It SHALL then set the NumAssocDev field to the number of entries in the NWKAddrAssocDevList field.

2.4.3.1.3 Node_Desc_req

The Node_Desc_req_command (ClusterID=0x0002) SHALL be formatted as illustrated in Figure 2-20.

Octets: 2	Octets: Variable
NWKAddrOfInterest	TLVs

Figure 2-20. Format of the Node_Desc_req Command Frame

Table 2-46 specifies the fields for the Node_Desc_req command frame.

Table 2-46. Fields of the Node_Desc_req Command Frame

Name	Type	Valid Range	Description
NWKAddrOfInterest	Device Address	16-bit NWK address	NWK address for the request
TLVs	Concatenation of TLVs	Varies	The Fragmentation Parameters Global TLV SHALL always be included. If the Node_Desc_req is sent to the Trust Center from a device wishing to update its Trust Center link-key, the Supported Key Negotiation Methods Global TLV (ID 65) SHALL be included.

2.4.3.1.3.1 When Generated

The Node_Desc_req command is generated from a local device wishing to inquire as to the node descriptor of a remote device. This command SHALL be unicast either to the remote device itself or to an alternative device that contains the discovery information of the remote device.

The local device SHALL generate the Node_Desc_req command using the format illustrated in . The NWKAddrOfInterest field SHALL contain the network address of the remote device for which the node descriptor is required.

The Fragmentation Parameters Global TLV SHALL be present to indicate the sending device's fragmentation capabilities. This allows the receiving device to cache the information if it needs to.

If the Node_Desc_req is sent to the Trust Center from a device wishing to update its Trust Center link-key, the Supported Key Negotiation Methods Global TLV (ID 65) SHALL be included.

2.4.3.1.3.2 Effect on Receipt

Upon receipt of this command, the recipient device SHALL process the command and generate a Node_Desc_rsp command in response, according to the description in section 2.4.4.2.3.

2.4.3.1.4 Power_Desc_req

The Power_Desc_req command (ClusterID=0x0003) SHALL be formatted as illustrated in Figure 2-21.

Octets: 2
NWKAddrOfInterest

Figure 2-21. Format of the Power_Desc_req Command Frame

Table 2-47 specifies the fields of the Power_Desc_req command frame.

Table 2-47. Fields of the Power_Desc_req Command Frame

Name	Type	Valid Range	Description
NWKAddrOfInterest	Device Address	16-bit NWK address	NWK address for the request.

2.4.3.1.4.1 When Generated

The Power_Desc_req command is generated from a local device wishing to inquire as to the power descriptor of a remote device. This command SHALL be unicast either to the remote device itself or to an alternative device that contains the discovery information of the remote device.

The local device SHALL generate the Power_Desc_req command using the format illustrated in Table 2-47. The NWKAddrOfInterest field SHALL contain the network address of the remote device for which the power descriptor is required.

2.4.3.1.4.2 Effect on Receipt

Upon receipt of this command, the recipient device SHALL process the command and generate a Power_Desc_rsp command in response according to the description in section 2.4.4.2.4.

2.4.3.1.5 Simple_Desc_req

The Simple_Desc_req command (ClusterID=0x0004) SHALL be formatted as illustrated in Figure 2-22.

Octets: 2	1
NWKAddrOfInterest	EndPoint

Figure 2-22. Format of the Simple_Desc_req Command Frame

Table 2-48 specifies the fields of the Simple_Desc_req command frame.

Table 2-48. Fields of the Simple_Desc_req Command

Name	Type	Valid Range	Description
NWKAddrOfInterest	Device Address	16-bit NWK address	NWK address for the request
Endpoint	8 bits	1–254	The endpoint on the destination

2.4.3.1.5.1 When Generated

The Simple_Desc_req command is generated from a local device wishing to inquire as to the simple descriptor of a remote device on a specified endpoint. This command SHALL be unicast either to the remote device itself or to an alternative device that contains the discovery information of the remote device.

The local device SHALL generate the Simple_Desc_req command using the format illustrated in Table 2-48. The NWKAddrOfInterest field SHALL contain the network address of the remote device for which the simple descriptor is required and the endpoint field SHALL contain the endpoint identifier from which to obtain the required simple descriptor.

2.4.3.1.5.2 Effect on Receipt

Upon receipt of this command, the recipient device SHALL process the command and generate a Simple_Desc_rsp command in response, according to the description in section 2.4.4.2.5.

2.4.3.1.6 Active_EP_req

The Active_EP_req command (ClusterID=0x0005) SHALL be formatted as illustrated in Figure 2-23.

Octets: 2
NWKAddrOfInterest

Figure 2-23. Format of the Active_EP_req Command Frame

Table 2-49 specifies the fields of the Active_EP_req command frame.

Table 2-49. Fields of the Active_EP_req Command

Name	Type	Valid Range	Description
NWKAddrOfInterest	Device Address	16-bit NWK address	NWK address for the request.

2.4.3.1.6.1 When Generated

The Active_EP_req command is generated from a local device wishing to acquire the list of endpoints on a remote device with simple descriptors. This command SHALL be unicast either to the remote device itself or to an alternative device that contains the discovery information of the remote device.

The local device SHALL generate the Active_EP_req command using the format illustrated in . The NWKAddrOfInterest field SHALL contain the network address of the remote device for which the active endpoint list is required.

2.4.3.1.6.2 Effect on Receipt

Upon receipt of this command, the recipient device SHALL process the command and generate an Active_EP_rsp command in response, according to the description in section 2.4.4.2.6.

2.4.3.1.7 Match_Desc_req

The Match_Desc_req command (ClusterID=0x0006) SHALL be formatted as illustrated in Figure 2-24.

Octets: 2	2	1	Variable	1	Variable
NWKAddrOfInterest	ProfileID	NumInClusters	InClusterList	NumOutClusters	OutClusterList

Figure 2-24. Format of the Match_Desc_req Command Frame

Table 2-50 specifies the fields of the Match_Desc_req command frame.

Table 2-50. Fields of the Match_Desc_req Command

Name	Type	Valid Range	Description
NWKAddrOfInterest	Device Address	16-bit NWK address	NWK address for the request.
ProfileID	Integer	0x0000 – 0xffff	Profile ID to be matched at the destination.
NumInClusters	Integer	0x00 – 0xff	The number of Input Clusters provided for matching within the InClusterList.
InClusterList	2 bytes * NumInClusters		List of Input ClusterIDs to be used for matching; the InClusterList is the desired list to be matched by the Remote Device (the elements of the InClusterList are the supported output clusters of the Local Device).
NumOutClusters	Integer	0x00 – 0xff	The number of Output Clusters provided for matching within OutClusterList.
OutClusterList	2 bytes * NumOutClusters		List of Output ClusterIDs to be used for matching; the OutClusterList is the desired list to be matched by the Remote Device (the elements of the OutClusterList are the supported input clusters of the Local Device).

2.4.3.1.7.1 When Generated

The Match_Desc_req command is generated from a local device wishing to find remote devices supporting a specific simple descriptor match criterion. This command SHALL either be broadcast to all devices for which macRxOnWhenIdle = TRUE, or unicast. If the command is unicast, it SHALL be directed either to the remote device itself or to an alternative device that contains the discovery information of the remote device.

The local device SHALL generate the Match_Desc_req command using the format illustrated in . The NWKAddrOfInterest field SHALL contain the network address indicating a broadcast to all devices for which macRxOnWhenIdle = TRUE (0xfffd) if the command is to be broadcast, or the network address of the remote device for which the match is required.

The remaining fields SHALL contain the required criterion for which the simple descriptor match is requested. The ProfileID field SHALL contain the identifier of the profile for which the match is being sought or the wildcard profile ID of 0xFFFF.

The NumInClusters field SHALL contain the number of elements in the InClusterList field. If the value of this field is 0, the InClusterList field SHALL NOT be included. If the value of the NumInClusters field is not equal to 0, the InClusterList field SHALL contain the list of input cluster identifiers for which the match is being sought.

The NumOutClusters field SHALL contain the number of elements in the OutClusterList field. If the value of this field is 0, the OutClusterList field SHALL NOT be included. If the value of the NumOutClusters field is not equal to 0, the OutClusterList field SHALL contain the list of output cluster identifiers for which the match is being sought.

2.4.3.1.7.2 Effect on Receipt

Upon receipt of this command, the recipient device SHALL process the command and generate a Match_Desc_rsp command in response, according to the description in section 2.4.4.2.7.

2.4.3.1.8 Complex_Desc_req – DEPRECATED

2.4.3.1.9 User_Desc_req – DEPRECATED

2.4.3.1.10 Discovery_Cache_req – DEPRECATED

2.4.3.1.11 Device_annce

The Device_annce command (ClusterID=0x0013) SHALL be formatted as illustrated in Figure 2-25.

Octets: 2	8	1
NWKAddr	IEEEAddr	Capability

Figure 2-25. Format of the Device_annce Command Frame

Table 2-51 specifies the fields of the Device_annce command frame.

Table 2-51. Fields of the Device_annce Command

Name	Type	Valid Range	Description
NWKAddr	Device Address	16-bit NWK address	NWK address for the Local Device
IEEEAddr	Device Address	64-bit IEEE address	IEEE address for the Local Device
Capability	Bitmap	See Figure 2-16.	Capability of the local device

2.4.3.1.11.1 When Generated

The Device_annce is provided to enable Zigbee devices on the network to notify other Zigbee devices that the device has joined or re-joined the network, identifying the device's 64-bit IEEE address and new 16-bit NWK address, and informing the Remote Devices of the capability of the Zigbee device. This command SHALL be invoked for all Zigbee end devices upon join or rejoin. This command MAY also be invoked by Zigbee routers upon join or rejoin as part of NWK address conflict resolution. The destination addressing on this primitive is broadcast to all devices for which macRxOnWhenIdle = TRUE.

2.4.3.1.11.2 Effect on Receipt

Routers and Coordinators SHALL first determine whether there is an address conflict with any other device on the network. End Devices are not required to detect address conflicts.

Address conflicts SHALL be determined as follows:

1. Using the value of the IEEEAddr in the received ZDO message examine NIB tables for the nwkAddressMap and nwkNeighborTable for a matching IEEE Address.
2. If a match is found AND that match has a different node ID than the value for NWKAddr in the received ZDO message then an address conflict has conflict has occurred. Do the following:
    a. If the conflicted entry is the nwkNeighborTable of the NIB AND the entry has a Relationship of 0x06, neighbor is a lost child, this indicates a local end device child has NOT changed parents and needs a new address. Perform an NLME-SET.req as follows.
        i. Set the corresponding entry in the nwkNeighborTable to have a Relationship field of 0x07, neighbor is a child that needs new address.
    b. Follow the procedure in section 3.6.1.10.3 to resolve the address conflict.
        i. No further processing of the message SHALL be done.

When no conflict is detected, all device types SHALL continue processing the ZDO device announce as indicated below.

Upon receipt, the Remote Device SHALL use the IEEEAddr in the message to find a match with any other IEEE address held in the Remote Device. If a match is detected, the Remote Device SHALL update the nwkAddressMap attribute of the NIB with the updated NWKAddr corresponding to the IEEEAddr received.

The Remote Device SHALL also use the NWKAddr in the message to find a match with any other 16-bit NWK address held in the Remote Device, even if the IEEEAddr field in the message carries the value of 0xffffffffffffffff. If a match is detected for a device with an IEEE address other than that indicated in the IEEEAddr field received, then this entry SHALL be marked as not having a known valid 16-bit NWK address.

2.4.3.1.12 Parent_annce

The Parent_annce command (ClusterID = 0x001F) SHALL be formatted as illustrated in Figure 2-26.

Octets: 1	**Variable**	**…**	**Variable**
NumberOfChildren	ChildInfo[0]	…	ChildInfo[n]

Figure 2-26. Format of the Parent Annce Message

Table 2-52 specifies the contents of the ChildInfo structure.

Table 2-52. Format of the ChildInfo Structure

Name	Type	Description
Extended Address	64-bit IEEE address	The IEEE address of the child bound to the parent.

2.4.3.1.12.1 When Generated

The Parent_annce is provided to enable Zigbee routers (including the coordinator) on the network to notify other Zigbee routers about all the end devices known to the local device. This command provides a means to resolve conflicts more quickly than aging out the child, when multiple routers purport to be the active parent of a particular end-device. The command MAY be broadcast from one router to all routers and the coordinator using the broadcast address 0xFFFC or unicast from one router to another router.

This message SHALL be generated if all the following conditions are met:

1. The router or coordinator device has rebooted.
2. The router or coordinator is operating in the joined state.

The message generated under the above circumstances SHALL be broadcast. Before broadcasting a Parent_annce message, the device SHALL start a countdown timer, *apsParentAnnounceTimer* equal to apsParentAnnounceBaseTimer + a random value from 0 to apsParentAnnounceJitterMax.

When the timer expires, a router SHALL examine its neighbor table for all devices. The router SHALL construct, but not yet send, an empty Parent_annce message and set NumberOfChildren to 0. For each end device in the neighbor table, it SHALL do the following.

1. If the Neighbor Table entry indicates a Device Type not equal to End Device (0x02), do not process this entry. Continue to the next one.
2. Incorporate end device information into the Parent_annce message by doing the following:
    a. Append a ChildInfo structure to the message.
    b. Increment NumberOfChildren by 1.
3. Note: The value of Keepalive Received for the Neighbor Table Entry is not considered.

After processing all entries in the neighbor table, if the NumberOfChildren is greater than 0, then it SHALL send the message to the all routers broadcast address (0xFFFC). If NumberOfChildren is 0, it SHALL discard the previously constructed Parent_annce message and not send it.

If the device has more ChildInfo entries than fit in a single message, it SHALL send additional messages. Each additional message needed SHALL trigger the device to calculate and start a new apsParentAnnounceTimer equal to apsParentAnnounceBaseTimer + a random value from 0 to apsParentAnnounceJitterMax. The local device SHALL wait until that timer expires before sending each additional message. The NumberOfChildren for each message shall be set according to the number of ChildInfo entries contained within the message.

If the device shall send multiple Parent_annce messages but receives a keepalive from an end device before it has sent the Parent_Annce message, it SHALL NOT include that device in the message.

2.4.3.1.12.2 Effect on receipt

If the message is received by an end device, it SHALL be dropped. No further processing SHALL be done.

Upon receipt of a broadcast Parent_annce, if the local device has a non-zero value for its apsParentAnnounceTimer it SHALL immediately re-calculate a new value and start a new countdown. The apsParentAnnounceTimer SHALL be set to apsParentAnnounceBaseTimer + a random value from 0 to apsParentAnnounceJitterMax. It SHALL continue processing the message.

A router SHALL construct, but not yet send, an empty Parent_Annce_rsp message with NumberOfChildren set to 0. It SHALL examine each Extended Address present in the message and search its Neighbor Table for an Extended Address entry that matches. For each match, process as follows:

1. If the Device Type is Zigbee End Device (0x02) and the Keepalive Received value is TRUE, do the following:
    a. It SHALL append to the Parent_annce_rsp frame the ChildInfo structure.
    b. Increment the NumberOfChildren by 1.
2. If the Device Type is not Zigbee End Device (0x02) or the Keepalive Received value is FALSE, do not process any further. Continue to the next entry.

If the NumberOfChildren field value is 0, the local device SHALL discard the previously constructed Parent_Annce_rsp. No response message SHALL be sent.

If the NumberOfChildren field in the Parent_Annce_rsp is greater than 0, it SHALL unicast the message to the sender of the Parent_Annce message.

If the device has more ChildInfo entries than fit in a single message, it SHALL send additional messages. These messages do not have to be jittered or delayed since they are unicast to a single device. Each Parent_annce_rsp SHALL set the NumberOfChildren field to the number of entries contained within the message.

2.4.3.1.13 User_Desc_set – DEPRECATED

2.4.3.1.14 System_Server_Discovery_req

The System_Server_Discovery_req command (ClusterID=0x0015) SHALL be formatted as illustrated in Figure 2-27.

Octets: 2
ServerMask

Figure 2-27. Format of the System_Server_Discovery_req Command Frame

Table 2-53 specifies the fields of the System_Server_Discovery_req command frame.

Table 2-53. Fields of the System_Server_Discovery_req Command Frame

Name	Type	Valid Range	Description
ServerMask	Bitmap	16 bits	See Table 2-34 for bit assignments.

2.4.3.1.14.1 When Generated

The System_Server_Discovery_req is generated from a Local Device wishing to discover the location of a particular system server or servers as indicated by the ServerMask parameter. The destination addressing on this request is 'broadcast to all devices for which macRxOnWhenIdle = TRUE.'

2.4.3.1.14.2 Effect on Receipt

Upon receipt, remote devices SHALL compare the ServerMask parameter to the Server Mask field in their own Node descriptor. If no bits are found to match, no action is taken. If any matching bits are found, the remote device SHALL send a System_Server_Discovery_rsp back to the originator using unicast transmission (with acknowledgement request) and indicating the matching bits.

2.4.3.1.15 Discovery_store_req – DEPRECATED

2.4.3.1.16 Node_Desc_store_req – DEPRECATED

2.4.3.1.17 Power_Desc_store_req – DEPRECATED

2.4.3.1.18 Active_EP_store_req – DEPRECATED

2.4.3.1.19 Simple_Desc_store_req – DEPRECATED

2.4.3.1.20 Remove_node_cache_req – DEPRECATED

2.4.3.1.21 Find_node_cache_req – DEPRECATED

2.4.3.1.22 Extended_Simple_Desc_req – DEPRECATED

2.4.3.1.23 Extended_Active_EP_req – DEPRECATED

2.4.3.2 Bind, Unbind, and Bind Management Client Services Primitives

Table 2-54 lists the primitives supported by Device Profile: Bind and Unbind Client Services. Each of these commands will be discussed in the following sections.

Table 2-54. Bind, Unbind, and Bind Management Client Service Commands

Bind and Unbind Client Services	Cluster ID	Client Transmission	Server Processing	Restricted Mode Only
End_Device_Bind_req	0x0020	Deprecated	Deprecated	-
Bind_req	0x0021	O	O	Yes
Unbind_req	0x0022	O	O	Yes
Bind_Register_req	0x0023	Deprecated	Deprecated	-
Replace_Device_req	0x0024	Deprecated	Deprecated	-
Store_Bkup_Bind_Entry_req	0x0025	Deprecated	Deprecated	-
Remove_Bkup_Bind_Entry_req	0x0026	Deprecated	Deprecated	-
Backup_Bind_Table_req	0x0027	Deprecated	Deprecated	-
Recover_Bind_Table_req	0x0028	Deprecated	Deprecated	-
Backup_Source_Bind_req	0x0029	Deprecated	Deprecated	-
Recover_Source_Bind_req	0x002a	Deprecated	Deprecated	-
Clear_All_Bindings_req	0x002b	O	M / O *	Yes

*The Clear_All_Bindings is optional if no binding table is present. If a Binding Table is supported then the Clear_All_Bindings_req command server processing is mandatory.

2.4.3.2.1 End_Device_Bind_req – DEPRECATED

2.4.3.2.2 Bind_req

The Bind_req command (ClusterID=0x0021) SHALL be formatted as illustrated in Figure 2-28.

Octets: 8	1	2	1	2/8	0/1
SrcAddress	SrcEndp	ClusterID	DstAddrMode	DstAddress	DstEndp

Figure 2-28. Format of the Bind_req Command Frame

Table 2-55 specifies the fields of the Bind_req command frame.

3358 **Table 2-55. Fields of the Bind_req Command**

Name	Type	Valid Range	Description
SrcAddress	IEEE Address	A valid 64-bit IEEE address	The IEEE address for the source.
SrcEndp	Integer	0x01 – 0xfe	The source endpoint for the binding entry.
ClusterID	Integer	0x0000 – 0xffff	The identifier of the cluster on the source device that is bound to the destination.
DstAddrMode	Integer	0x00 – 0xff	The addressing mode for the destination address used in this command. This field can take one of the non-reserved values from the following list: 0x00 = reserved 0x01 = 16-bit group address for DstAddress and DstEndp not present 0x02 = reserved 0x03 = 64-bit extended address for DstAddress and DstEndp present 0x04 – 0xff = reserved
DstAddress	Address	As specified by the DstAddr-Mode field	The destination address for the binding entry.
DstEndp	Integer	0x01 – 0xfe	This field SHALL be present only if the DstAddr-Mode field has a value of 0x03 and, if present, SHALL be the destination endpoint for the binding entry.

3359 #### 2.4.3.2.2.1 When Generated

3360 The Bind_req is generated from a Local Device wishing to create a Binding Table entry for the source and destination
3361 addresses contained as parameters. The destination addressing on this command SHALL be unicast only, and the
3362 destination address SHALL be that of the SrcAddress itself. The Binding Manager is optionally supported on the
3363 source device (unless that device is also the Zigbee Coordinator) so that device SHALL issue a NOT_SUPPORTED
3364 status to the Bind_req if not supported.

3365 #### 2.4.3.2.2.2 Effect on Receipt

3366 On receipt of a broadcast Bind request the stack SHALL drop the message and no further processing SHALL take
3367 place. Otherwise, upon receipt, a Remote Device SHALL create a Binding Table entry based on the parameters sup-
3368 plied in the Bind_req if the Binding Manager is supported. The Remote Device SHALL then respond with SUCCESS
3369 if the entry has been created by the Binding Manager; otherwise, the Remote Device SHALL respond with INSUF-
3370 FICIENT_SPACE.

3371 ### 2.4.3.2.3 Unbind_req

3372 The Unbind_req command (ClusterID=0x0022) SHALL be formatted as illustrated in Figure 2-29.

Octets: 8	1	2	1	2/8	0/1
SrcAddress	SrcEndp	ClusterID	DstAddrMode	DstAddress	DstEndp

Figure 2-29. Format of the Unbind_req Command Frame

Table 2-56 specifies the fields of the Unbind_req command frame.

Table 2-56. Fields of the Unbind_req Command

Name	Type	Valid Range	Description
SrcAddress	IEEE Address	A valid 64-bit IEEE address	The IEEE address for the source
SrcEndp	Integer	0x01 – 0xfe	The source endpoint for the binding entry
ClusterID	Integer	0x0000 – 0xffff	The identifier of the cluster on the source device that is bound to the destination.
DstAddrMode	Integer	0x00 – 0xff	The addressing mode for the destination address used in this command. This field can take one of the non-reserved values from the following list: 0x00 = reserved 0x01 = 16-bit group address for DstAddress and DstEndp not present 0x02 = reserved 0x03 = 64-bit extended address for DstAddress and DstEndp present 0x04 – 0xff = reserved
DstAddress	Address	As specified by the DstAddrMode field	The destination address for the binding entry.
DstEndp	Integer	0x01 – 00xfe	This field SHALL be present only if the DstAddrMode field has a value of 0x03 and, if present, SHALL be the destination endpoint for the binding entry.

2.4.3.2.3.1 When Generated

The Unbind_req is generated from a Local Device wishing to remove a Binding Table entry for the source and destination addresses contained as parameters. The destination addressing on this command SHALL be unicast only and the destination address SHALL be that of the SrcAddress.

2.4.3.2.3.2 Effect on Receipt

On receipt of a broadcast Unbind request the stack SHALL drop the message and no further processing SHALL be done. The Remote Device SHALL evaluate whether this request is supported. If the request is not supported, a Status of NOT_SUPPORTED SHALL be returned. If the request is supported, the Remote Device SHALL remove a Binding Table entry based on the parameters supplied in the Unbind_req. If a Binding Table entry for the SrcAddress, SrcEndp, ClusterID, DstAddress, DstEndp contained as parameters does not exist, the Remote Device SHALL respond with

NO_ENTRY. Otherwise, the Remote Device SHALL delete the indicated Binding Table entry and respond with SUCCESS.

2.4.3.2.4 Bind_Register_req – DEPRECATED

2.4.3.2.5 Replace_Device_req – DEPRECATED

2.4.3.2.6 Store_Bkup_Bind_Entry_req – DEPRECATED

2.4.3.2.7 Remove_Bkup_Bind_Entry_req – DEPRECATED

2.4.3.2.8 Backup_Bind_Table_req – DEPRECATED

2.4.3.2.9 Recover_Bind_Table_req – DEPRECATED

2.4.3.2.10 Backup_Source_Bind_req – DEPRECATED

2.4.3.2.11 Recover_Source_Bind_req – DEPRECATED

2.4.3.2.12 Clear_All_Bindings_req

The Clear_All_Bindings_req command (Cluster = 0x002b) SHALL be formatted as described in Figure 2-30. Any device on the network can send this command subject to the same Restricted Mode processing rules that apply to other commands manipulating the binding table.

Octets: Varies
TLVs

Figure 2-30. Format of the Clear_All_Bindings_req

The following TLVs SHALL be present in the message:

- Clear All Bindings Req EUI64 TLV

2.4.3.2.12.1 Local TLVs

2.4.3.2.12.2 Clear All Bindings Req EUI64 TLV (ID=0)

The format of the Clear All Bindings Req EUI64 TLV SHALL be as formatted in Figure 2-31.

Octets: 1	8	…
EUI64 Count	EUI64	…

Figure 2-31. Format of the Clear All Bindings Req EUI64 TLV

The fields of the Clear All Bindings Req EUI64 TLV are defined in Table 2-57.

3409 **Table 2-57. Fields of the Clear All Bindings Req EUI64 TLV**

Name	Type	Valid Range	Description
EUI64 Count	Integer	0x00 – 0xFF	The number of EUI64 fields within the TLV. NOTE: The Maximum Transmission Unit (MTU) of the underlying message will limit the maximum range of this field.
EUI64	EUI64	0x0000000000000000 – 0xFFFFFFFFFFFFFFFF	An EUI64 that SHALL trigger corresponding bindings to be deleted.

3410 2.4.3.2.12.3 **When Generated**

3411 This is generated by a remote device that wants to clear all the bindings of the local device, for example to clear the
3412 application configuration without resetting the device to its factory defaults and causing it to drop off the network.
3413 This command SHALL be sent via unicast.

3414 2.4.3.2.12.4 **Effect on Receipt**

3415 The receiver SHALL do the following:

3416 1) If the command was broadcast, the command SHALL be dropped and no further processing SHALL be done.
3417 2) Perform TLV processing rules as described in Annex I (General TLV Processing section).
3418 3) If the command does not include a Clear All Bindings Req EUI64 TLV in the message, then it SHALL be re-
3419 jected.
3420 a) A ZDO Clear_All_Bindings_rsp SHALL be generated with a status of INV_REQUESTTYPE. No further
3421 processing SHALL be done to clear the application configuration without resetting the device to its factory
3422 defaults and causing it to drop off the network.
3423 4) For each EUI64 in the Clear All Bindings Req EUI64 TLV search the Binding Table and delete any binding
3424 that matches that EUI64. If the Wildcard EUI64 of 0xFFFFFFFFFFFFFFFF is used then all bindings on the lo-
3425 cal device SHALL be deleted.
3426 5) Generate a ZDO Clear_All_Bindings_rsp containing a Status Field.
3427 a) Set the status to SUCCESS

3428 2.4.3.3 Network Management Client Services

3429 Table 2-58 lists the commands supported by Device Profile: Network Management Client Services. Each of these
3430 primitives will be discussed in the following sections.

3431 **Table 2-58. Network Management Client Services Commands**

Network Management Client Services	Cluster ID	Client Transmission	Server Processing	Restricted Command
Mgmt_NWK_Disc_req	0x0030	Deprecated	Deprecated	-
Mgmt_Lqi_req	0x0031	O	M	No
Mgmt_Rtg_req	0x0032	O	M	No
Mgmt_Bind_req	0x0033	O	M	No

Network Management Client Services	Cluster ID	Client Transmission	Server Processing	Restricted Command
Mgmt_Leave_req	0x0034	O	M	**Yes**
Mgmt_Direct_Join_req	0x0035	Deprecated	Deprecated	-
Mgmt_Permit_Joining_req	0x0036	O	M	No
Mgmt_Cache_req	0x0037	Deprecated	Deprecated	-
Mgmt_NWK_Update_req	0x0038	O	O	No
Mgmt_NWK_Enhanced_Update_req	0x0039	O	O	No
Mgmt_NWK_IEEE_Joining_List_req	0x003a	O	O	No
Reserved	0x003b	-	-	-
Mgmt_NWK_Beacon_Survey_req	0x003c	O	M*	No

* The Mgmt_NWK_Beacon_Survey_req server processing is mandatory for End Devices and optional for routers.

2.4.3.3.1 Mgmt_NWK_Disc_req – DEPRECATED

2.4.3.3.2 Mgmt_Lqi_req

The Mgmt_Lqi_req command (ClusterID=0x0031) SHALL be formatted as illustrated in Figure 2-32.

Octets: 1
StartIndex

Figure 2-32. Format of the Mgmt_Lqi_req Command Frame

Table 2-59 specifies the fields for the Mgmt_NWK_Disc_req command frame.

Table 2-59. Fields of the Mgmt_Lqi_req Command

Name	Type	Valid Range	Description
StartIndex	Integer	0x00 – 0xff	Starting Index for the requested elements of the Neighbor Table.

2.4.3.3.2.1 When Generated

The Mgmt_Lqi_req is generated from a Local Device wishing to obtain a neighbor list for the Remote Device along with associated LQA values to each neighbor. The destination addressing on this command SHALL be unicast only. It MAY be sent to a coordinator, router, or end device.

2.4.3.3.2.2 Effect on Receipt

Upon receipt, a Remote Device (Zigbee Router or Zigbee Coordinator) SHALL retrieve the entries of the neighbor table and associated LQA values via the NLME-GET.request primitive (for the *nwkNeighborTable* attribute) and report the resulting neighbor table (obtained via the NLME-GET.confirm primitive) via the Mgmt_Lqi_rsp command.[2]

Prior to Revision 21 of this specification, server processing of this command was optional. Additionally end devices were not required to support the command. As a result some devices MAY return NOT_SUPPORTED. For R22 and beyond, all devices SHALL support this command.

Prior to Revision 23 of this specification, the LQI value was returned, which might have exhibited more platform-specific behavior.

If this command is not supported in the Remote Device, the return status provided with the Mgmt_Lqi_rsp SHALL be NOT_SUPPORTED. If the neighbor table was obtained successfully, the Mgmt_Lqi_rsp command SHALL contain a status of SUCCESS and the neighbor table SHALL be reported, beginning with the element in the list enumerated as StartIndex. If the neighbor table was not obtained successfully, the Mgmt_Lqi_rsp command SHALL contain the error code reported in the NLME-GET.confirm primitive.

2.4.3.3.3 Mgmt_Rtg_req

The Mgmt_Rtg_req command (ClusterID=0x0032) SHALL be formatted as illustrated in Figure 2-33.

Octets: 1
StartIndex

Figure 2-33. Format of the Mgmt_Rtg_req Command Frame

Table 2-60 specifies the fields for the Mgmt_Rtg_req command frame.

Table 2-60. Fields of the Mgmt_Rtg_req Command

Name	Type	Valid Range	Description
StartIndex	Integer	0x00-0xff	Starting Index for the requested elements of the Routing Table.

2.4.3.3.3.1 When Generated

The Mgmt_Rtg_req is generated from a Local Device wishing to retrieve the contents of the Routing Table from the Remote Device. The destination addressing on this command SHALL be unicast only and the destination address SHALL be that of the Zigbee Router or Zigbee Coordinator.

[2] CCB 2265

2.4.3.3.3.2 Effect on Receipt

Upon receipt, a Remote Device (Zigbee Coordinator or Zigbee Router) SHALL retrieve the entries of the routing table from the NWK layer via the NLME-GET.request primitive (for the *nwkRouteTable* attribute) and report the resulting routing table (obtained via the NLME-GET.confirm primitive) via the Mgmt_Rtg_rsp command.

If the Remote Device does not support this optional management request, it SHALL return a Status of NOT_SUPPORTED. If the routing table was obtained successfully, the Mgmt_Rtg_req command SHALL contain a status of SUCCESS and the routing table SHALL be reported, beginning with the element in the list enumerated as StartIndex. If the routing table was not obtained successfully, the Mgmt_Rtg_rsp command SHALL contain the error code reported in the NLME-GET.confirm primitive.

2.4.3.3.4 Mgmt_Bind_req

The Mgmt_Bind_req command (ClusterID=0x0033) SHALL be formatted as illustrated in Figure 2-34.

Octets: 1
StartIndex

Figure 2-34. Format of the Mgmt_Bind_req Command Frame

Table 2-61 specifies the fields for the Mgmt_Bind_req command frame.

Table 2-61. Fields of the Mgmt_Bind_req Command

Name	Type	Valid Range	Description
StartIndex	Integer	0x00 – 0xff	Starting Index for the requested elements of the Binding Table.

2.4.3.3.4.1 When Generated

The Mgmt_Bind_req is generated from a Local Device wishing to retrieve the contents of the Binding Table from the Remote Device. The destination addressing on this command SHALL be unicast only and the destination address SHALL be that of a source device holding its own binding table.

2.4.3.3.4.2 Effect on Receipt

Upon receipt, a Remote Device SHALL retrieve the entries of the binding table from the APS sub-layer via the APSME-GET.request primitive (for the *apsBindingTable* attribute) and report the resulting binding table (obtained via the APSME-GET.confirm primitive) via the Mgmt_Bind_rsp command.

If the Remote Device does not support this optional management request, it SHALL return a status of NOT_SUPPORTED. If the binding table was obtained successfully, the Mgmt_Bind_rsp command SHALL contain a status of SUCCESS and the binding table SHALL be reported, beginning with the element in the list enumerated as StartIndex. If the binding table is empty, the Mgmt_Bind_rsp SHALL return SUCCESS, set the fields BindingTable Entries = Start Index = BindingTable ListCount = 0x00 and not include the BindingTable List field. If the binding table was not obtained successfully, the Mgmt_Bind_rsp command SHALL contain the error code reported in the APSME-GET.confirm primitive.

2.4.3.3.5 Mgmt_Leave_req

The Mgmt_Leave_req command (ClusterID=0x0034) SHALL be formatted as illustrated in Figure 2-35.

Bits: 64	6	1	1
Device Address	Reserved	Remove Children	Rejoin

Figure 2-35. Format of the Mgmt_Leave_req Command Frame

Table 2-62 specifies the fields for the Mgmt_Leave_req command frame.

Table 2-62. Fields of the Mgmt_Leave_req Command

Name	Type	Valid Range	Description
DeviceAddress	Device Address	An extended 64-bit, IEEE address	See section 3.2.2.18 for details on the Device Address parameter within NLME-LEAVE.request. For DeviceAddress of NULL, a value of 0x0000000000000000 SHALL be used.
Remove Children	Bit	0 or 1	This field has a value of 1 if the device being asked to leave the network is also being asked to remove its child devices, if any. Otherwise, it has a value of 0.
Rejoin	Bit	0 or 1	This field has a value of 1 if the device being asked to leave from the current parent is requested to rejoin the network. Otherwise, it has a value of 0.

2.4.3.3.5.1 When Generated

The Mgmt_Leave_req is generated from a Local Device requesting that a Remote Device leave the network or to request that another device leave the network. The Mgmt_Leave_req is generated by a management application which directs the request to a Remote Device where the NLME-LEAVE.request is to be executed using the parameter supplied by Mgmt_Leave_req.

2.4.3.3.5.2 Effect on Receipt

Upon receipt, the remote device SHALL process the leave request by executing the procedure in section 3.6.1.11.3.1. If the leave request was validated and accepted, and the DeviceAddress in the request is equal to the local device's EUI64, then the receiving device SHALL generate the NLME-LEAVE.request to disassociate from the currently associated network. The NLME-LEAVE.request SHALL have the DeviceAddress parameter set to the local device's *nwkIeeeAddress* from the NIB, the RemoveChildren SHALL be set to FALSE, and the Rejoin parameter SHALL be set to FALSE.

The results of the leave attempt SHALL be reported back to the local device via the Mgmt_Leave_rsp command. If the request was for the local device, then the Mgmt_Leave_rsp SHALL be sent prior to leaving the network.

Versions of this specification prior to Revision 21 did not mandate the requirement to support this command. Therefore if the remote device did not support this optional management request, it would return a status of NOT_SUPPORTED. All devices certified against version 21 and later are now required to support this command.

If the leave attempt was executed successfully, the Mgmt_Leave_rsp command SHALL contain a status of SUCCESS. If the leave attempt was not executed successfully, the Mgmt_Leave_rsp command SHALL contain the error code reported in the NLME-LEAVE.confirm primitive.

2.4.3.3.6 Mgmt_Direct_Join_req – DEPRECATED

2.4.3.3.7 Mgmt_Permit_Joining_req

The Mgmt_Permit_Joining_req command (ClusterID=0x0036) SHALL be formatted as illustrated in Figure 2-36.

Octets: 1	1	Variable
PermitDuration	TC_Significance	TLV Data

Figure 2-36. Format of the Mgmt_Permit_Joining_req Command Frame

Table 2-63 specifies the fields of the Mgmt_Permit_Joining_req command frame.

Table 2-63. Fields of the Mgmt_Permit_Joining_req Command

Name	Type	Valid Range	Description
PermitDuration	Integer	0x00 – 0xfe	See section 3.2.2.7 for details on the PermitDuration parameter within NLME-PERMIT-JOINING.request.
TC_Significance	Boolean Integer	0x00 – 0x01	This field SHALL always have a value of 1, indicating a request to change the Trust Center policy. If a frame is received with a value of 0, it shall be treated as having a value of 1.
TLV Data	Variable	Variable	This is a concatenated list of TLVs. This field was added in Revision 23.

2.4.3.3.7.1 When Generated

The Mgmt_Permit_Joining_req is generated from a Local Device requesting a change to the network's advertisement of its status, such as permitting joining. The Mgmt_Permit_Joining_req is generated by a management application or commissioning tool which directs the request to a remote device(s). Additionally, if the remote device is the Trust Center and TC_Significance is set to 1, this command is a request to change the Trust Center's policy to allow new devices to join. The Trust Center has the ultimate decision over whether this request will be accepted. The addressing MAY be unicast or 'broadcast to all routers and coordinator.

Trust Centers are the only devices allowed to update the Zigbee Beacon Appendix data advertised by the network in the IEEE Std 802.15.4 beacons. The network wide Beacon Appendix data is stored in the NIB value *nwkNetworkWideBeaconAppendixTLVs*.

The Trust Center can modify the nwkNetworkWideBeaconAppendixTLVs of all routers by setting data in the Beacon Appendix Encapsulation Global TLV. At a minimum the Trust Center SHALL always include the Beacon Appendix Encapsulation Global TLV as a TLV in the TLV Data field of a Mgmt_Permit_Joining_req. This is regardless of the value it sets for the PermitDuration field. Inside the Beacon Appendix Encapsulation Global TLV SHALL be the following TLVs:

- Supported Key Negotiation Methods Global TLV
- Fragmentation Parameter Global TLV

The Trust Center can include additional Global TLVs in the encapsulation TLV. Local TLVs SHALL NOT be stored in the Beacon Appendix Encapsulation Global TLV. The *nwkNetworkWideBeaconAppendixTLVs* SHALL always be set in its entirety by the Beacon Appendix Encapsulation Global TLV and SHALL NOT be appended to. The *nwkNetworkWideBeaconAppendixTLVs* NIB value SHALL NOT be persisted across reboots.

Additional local or global TLVs MAY be included in the TLV Data field of the Mgmt_Permit_Joining_req alongside the Beacon Appendix Encapsulation Global TLV. These TLVs do not change the state of the *nwkNetwork-WideBeaconAppendixGlobalTLVs*.

Non-Trust Center devices are not allowed to change the network wide Beacon Appendix data advertised by the network, only the permit joining duration. Non-Trust Center devices initiating this message SHALL not include the Beacon Appendix Encapsulation Global TLV. They MAY include other TLVs in the TLV Data field of the Mgmt_Permit_Joining_req.

2.4.3.3.7.2 Effect on Receipt

Upon receipt, the remote device(s) SHALL issue the NLME-PERMIT-JOINING.request primitive using the PermitDuration parameter supplied with the Mgmt_Permit_Joining_req command. If the PermitDuration parameter is not equal to zero or 0xFF, the parameter is a number of seconds and joining is permitted until it counts down to zero, after which time, joining is not permitted. If the PermitDuration is set to zero, joining is not permitted. Versions of this specification prior to Revision 21 allowed a value of 0xFF to be interpreted as 'forever'. Version 21 and later do not allow this. All devices conforming to this specification SHALL interpret 0xFF as 0xFE Devices that wish to extend the PermitDuration beyond 0xFE seconds SHALL periodically re-send the Mgmt_Permit_Joining_req.

If a second Mgmt_Permit_Joining_req is received while the previous one is still counting down, it will supersede the previous request.

A value of zero for the TC_Significance field has been deprecated. The field SHALL always be included in the message and all received frames SHALL be treated as though set to 1, regardless of the actual received value. In other words, all Mgmt_Permit_Joining_req SHALL be treated as a request to change the TC Policy.

If the remote device is the Trust Center the Trust Center authorization policy MAY be affected. Whether the Trust Center accepts a change in its authorization policy is dependent upon its Trust Center policies. A Trust Center device receiving a Mgmt_Permit_Joining_req SHALL execute the procedure in section 4.7.3.4 to determine if the request is permitted. If the operation was not permitted, the status code of INV_REQUESTTYPE SHALL be set. If the operation was allowed, the status code of SUCCESS SHALL be set.

If the Mgmt_Permit_Joining_req primitive was received as a unicast, the results of the NLME-PERMIT-JOINING.request SHALL be reported back to the local device via the Mgmt_Permit_Joining_rsp command. If the command was received as a broadcast, no response SHALL be sent back.

Prior to Revision 23 the TLV Data was never present. With Revision 23 and beyond, the TLV Data field can be present depending on whether the message was initiated by a Revision 23 device. Devices prior to Revision 23 SHALL ignore the TLV Data field on receipt and will never transmit the message with this field present.

If the Beacon Appendix Encapsulation Global TLV is present the receiver SHALL store all Global TLVs from the TLV Data in the *nwkNetworkWideBeaconPayloadTLVs* of the NIB, the Beacon Appendix Encapsulation Global TLV container SHALL not be stored. If Local TLVs are stored inside the Beacon Appendix Encapsulation TLV they SHALL be discarded and not stored in the nwkNetworkWideBeaconPayloadTLVs. If the Beacon Appendix Encapsulation Global TLV Data has no data inside it, the receiver SHALL clear the contents of the *nwkNetwork-WideBeaconPayloadTLVs* of the NIB .If the Beacon Appendix Encapsulation Global TLV is not present, then no changes are made to the contents of the *nwkNetworkWideBeaconPayloadTLVs* of the NIB.

2.4.3.3.8 Mgmt_Cache_req – DEPRECATED

2.4.3.3.9 Mgmt_NWK_Update_req

This command only supports the 2.4 GHz channel list. For other channels, see the Mgmt_NWK_Enhanced_Update_req.

The Mgmt_NWK_Update_req command (ClusterID=0x0038) SHALL be formatted as illustrated in Figure 2-37.

Octets: 4	1	0/1	0/1	0/2
ScanChannels	ScanDuration	ScanCount	*nwkUpdateId*	*nwkManagerAddr*

Figure 2-37. Fields of the Mgmt_NWK_Update_req Command Frame

Table 2-64 specifies the fields of the Mgmt_NWK_Update_req command frame.

Table 2-64. Fields of the Mgmt_NWK_Update_req Command

Name	Type	Valid Range	Description
ScanChannels	Bitmap	32-bit field	See Table 3-7 for details on the 32-bit field structure..
ScanDuration	Integer	0x00 – 0x05 or 0xfe or 0xff	A value used to calculate the length of time to spend scanning each channel. The time spent scanning each channel is (aBaseSuperframeDuration * (2^n + 1)) symbols, where n is the value of the ScanDuration parameter. For more information on MAC sub-layer scanning (see [B1]). If ScanDuration has a value of 0xfe this is a request for channel change. If ScanDuration has a value of 0xff this is a request to change the *apsChannelMaskList* and *nwkManagerAddr* attributes.
ScanCount	Integer	0x00 – 0x01	This field represents the number of energy scans to be conducted and reported. This field SHALL be present only if the ScanDuration is within the range of 0x00 to 0x05.
nwkUpdateId	Integer	0x00 – 0xFF	The value of the *nwkUpdateId* contained in this request. This value is set by the Network Channel Manager prior to sending the message. This field SHALL only be present of the ScanDuration is 0xfe or 0xff. If the ScanDuration is 0xff, then the value in the *nwkUpdateID* SHALL be ignored.
nwkManagerAddr	Device Address	16-bit NWK address	This field SHALL be present only if the ScanDuration is set to 0xff, and, where present, indicates the NWK address for the device with the Network Manager bit set in its Node Descriptor.

2.4.3.3.9.1 When Generated

This command is provided to allow updating of network configuration parameters or to request information from devices on network conditions in the local operating environment. The destination addressing on this primitive SHALL be unicast or broadcast to all devices for which macRxOnWhenIdle = TRUE.

2.4.3.3.9.2 Effect on Receipt

This section applies to both Mgmt_NWK_ Update_req and Mgmt_NWK_Enhanced_Update_req.

If Mgmt_NWK_Enhanced_Update_req is received and the server for it is not present, the device SHALL respond with NOT_SUPPORTED.

This command can cause the remote device to update its channel mask and network manager address, perform a channel change, or execute a channel scan. Processing is as follows.

1) If the received message is Mgmt_NWK_Update_req, the local device SHALL construct a ChannelListStructure for page 0 from the ScanChannels bitmap.
    a) Continue processing.
2) If the received message is Mgmt_NWK_Enhanced_Update_req, the local device SHALL construct a ChannelListStructure from the ScanChannelsListStructure.
    a) Continue processing.
3) If the ScanDuration parameter is equal to 0xfe, the message is a command to change channels. The device SHALL do the following.
    a) If the nwkNextChannelChange value in the NIB is non-zero, do the following. Compare the channel to change received over the air to the value in the NIB. If the values do not match, do the following:
        i) Follow the Error Response procedure setting the status to NOT_AUTHORIZED.
        ii) The request SHALL be dropped and no more processing SHALL take place.
    b) If there is more than 1 channel indicated in the ScanChannels bitmap (if the message is Mgmt_NWK_Update_req) or in the ScanChannelsListStructure (if the message is Mgmt_NWK_Enhanced_Update_req), then this is an invalid request. Do the following:
        i) Follow the Error Response procedure setting the status to INV_REQUESTTYPE.
        ii) The request SHALL be dropped and no more processing SHALL take place.
    c) The receiving device SHALL determine if the channel is one within the range of all supported channels.
        i) Examine the SupportedChannels element for each entry in the nwkMacInterfaceTable, and determine if there is a match within the received ScanChannels bitmap or ScanChannelsListStructure.
        ii) If no match is found, do the following:
            (1) Follow the Error Response procedure setting the status to INV_REQUESTTYPE.
            (2) The request SHALL be dropped and no more processing SHALL take place.
        iii) If a match is found, perform a channel change.
            (1) Execute a MLME-SET.request for the PIB value phyCurrentPage.
            (2) Execute a MLME-SET.request for the PIB value phyCurrentChannel.
            (3) No further processing SHALL be done.
4) If the ScanDuration parameter is equal to 0xff, the command provides a new apsChannelMaskList along with a new nwkManagerAddr. The device SHALL do the following:
    a) If the *apsTrustCenterAddress* of the AIB is set to a value other than 0xFFFFFFFFFFFFFFFF (distributed security network) and the nwkManagerAddr in the request is not 0x0000, the request SHALL be dropped and no more processing SHALL be done.
    b) If the received command is Mgmt_NWK_Update_req, set the apsChannelMaskList in the AIB to the value of the ScanChannels bitmap in the request.
    c) If the received command is a Mgmt_NWK_Enhanced_Update_req, use the value of the ScanChannelsListStructure in the request , to update the apsChannelMaskList in the AIB.
    d) Execute an NMLE-SET.request setting the nwkManagerAddr in the NIB to the value of the nwkManagerAddr in the request.

e) No more processing shall be done.

5) If the ScanDuration parameter is between 0x00 and 0x05, it is a request to do a channel scan. The device SHALL do the following:

a) If the request was not unicast, the request SHALL be dropped and no more processing SHALL be done.

b) For each entry in the nwkMacInterfaceTable, examine the SupportedChannels element and determine if there is a match.

i) If no match is found, do the following:

(1) Follow the Error Response procedure setting in section 2.4.3.3.9.3.

(2) The request SHALL be dropped and no more processing SHALL be done.

c) If the request is a Mgmt_NWK_Enhanced_Update_req and the ScanChannelsListStructure includes more than one page, do the following:

i) Follow the Error Response procedure setting the status to INV_REQUESTTYPE.

ii) The request SHALL be dropped and no more processing SHALL be done.

d) If a match is found, perform an Energy Detect Scan on the requested channels. The following procedure SHALL be executed a number of times equal to the ScanCount.

i) Execute a MLME-SCAN.request as follows.

(a) ScanType SHALL be set to ENERGY.

(b) ScanChannels SHALL be set to the matching channels in the current page.

(c) ChannelPage SHALL be set to the current page.

(d) ScanDuration SHALL be set to the ScanDuration in the request.

ii) If the received message is a Mgmt_NWK_Update_req, on receipt of the MLME-SCAN.confirm, generate a Mgmt_NWK_Update_notify with the status of the MLME-SCAN.confirm.

iii) If the received message is a Mgmt_NWK_Enhanced_Update_req, on receipt of the MLME-SCAN.confirm, generate a Mgmt_NWK_Enhanced_Update_notify with the status of the MLME-SCAN.confirm..

6) If the ScanDuration is any other value, the device SHALL do the following.

a) Execute the Error Response Procedure setting the status to INV_REQUESTTYPE.

b) No further processing SHALL be done.

2.4.3.3.9.3 Error Response Procedure

If it is determined that the error response procedure SHALL be executed, the device SHALL do the following:

1) If the request was broadcast, no response SHALL be generated.

2) If the request was unicast, a response SHALL be generated as follows:

a) Set the status according to the result of the operation.

b) If the request was a Mgmt_NWK_Update_req, generate a Mgmt_NWK_Update_notify.

c) If the request was a Mgmt_NWK_Enhanced_Update_req, generate a Mgmt_NWK_Enhanced_Update_notify.

2.4.3.3.10 Mgmt_NWK_Enhanced_Update_req

The Mgmt_NWK_Enhanced_Update_req command (ClusterID=0x0039) SHALL be formatted as illustrated in Figure 2-38.

Variable	1	0/1	0/1	0/2	0/1
ScanChannel-ListStructure	ScanDuration	ScanCount	*nwkUpdateId*	*nwkManagerAddr*	ConfigurationBitmask

Figure 2-38. Fields of the Mgmt_NWK_Enhanced_Update_req

Table 2-65 specifies the fields of the Mgmt_NWK_Enhanced_Update_req command frame.

Table 2-65. Field Descriptions of the Mgmt_NWK_Enhanced_Update_req

Name	Type	Valid Range	Description
ScanChannelsListStructure	ChannelListStructure	Variable	The list of channels and pages over which the scan is to be done. For more information on the Channel List structure see section 3.2.2.2.1. If ScanDuration is in the range 0x00 to 0x05, this parameter SHALL be restricted to a single page.
ScanDuration	Integer	0x00 – 0x05 or 0xfe or 0xff	A value used to calculate the length of time to spend scanning each channel. The time spent scanning each channel is (aBaseSuperframeDuration * (2^n + 1)) symbols, where n is the value of the ScanDuration parameter. For more information on MAC sub-layer scanning (see [B1]). If ScanDuration has a value of 0xfe this is a request for channel change. If ScanDuration has a value of 0xff this is a request to change the *apsChannelMaskList* and *nwkManagerAddr* attributes.
ScanCount	Integer	0x00 – 0x01	This field represents the number of energy scans to be conducted and reported. This field SHALL be present only if the ScanDuration is within the range of 0x00 to 0x05.
nwkUpdateId	Integer	0x00 – 0xFF	The value of the *nwkUpdateId* contained in this request. This value is set by the Network Channel Manager prior to sending the message. This field SHALL only be present if the ScanDuration is 0xfe or 0xff.

Name	Type	Valid Range	Description
			If the ScanDuration is 0xff, then the value in the *nwkUpdateID* SHALL be ignored.
nwkManagerAddr	Device Address	16-bit NWK address	This field SHALL be present only if the ScanDuration is set to 0xff, and, where present, indicates the NWK address for the device with the Network Manager bit set in its Node Descriptor.
ConfigurationBitmask			Defined in defined in section 2.4.3.3.12. The configurationBitmask must be added to the end of the list of parameters. This octet may or may not be present. If not present then assumption should be that it is enhanced active scan. If present then the configuration bitmask shall indicate the type of scan required.

2.4.3.3.10.1 When Generated

This command is provided to allow updating of network configuration parameters or to request information from devices on network conditions in the local operating environment. The destination addressing on this primitive SHALL be unicast or broadcast to all devices for which macRxOnWhenIdle = TRUE.

2.4.3.3.10.2 Effect on Receipt

Follow the procedure in .

2.4.3.3.11 Mgmt_NWK_IEEE_Joining_List_req

The Mgmt_NWK_IEEE_Joining_List_req command is provided as a mechanism to obtain the list of IEEE addresses that are EXPECTED to be joining the network. This allows the local router to filter Enhanced Beacon Requests and only respond to the devices that are joining.

The Mgmt_NWK_IEEE_Joining_List_req (Cluster ID 0x003A) command SHALL be formatted as illustrated in Figure 2-39.

Octets: 1
StartIndex

Figure 2-39. Fields of the Mgmt_NWK_IEEE_Joining_List_req

Table 2-66 describes the fields of the Mgmt_NWK_IEEE_Joining_List_req command.

Table 2-66. Field Descriptions of the Mgmt_NWK_IEEE_Joining_List_req

Name	Type	Valid Range	Description
StartIndex	Integer	0x00 – 0xFF	The starting index into the receiving device's nwkIeeeJoiningList that SHALL be sent back.

2.4.3.3.11.1 When Generated

The Mgmt_NWK_IEEE_Joining_List_req is generated from a local device requesting to get the mibJoinPolicyTable after being authenticated on the network.

2.4.3.3.11.2 Effect on Receipt

This command was introduced in R22 of this specification. It is mandatory for all Coordinator and Router devices to implement this going forward but older stack versions SHALL return a ZDO Status of NOT_SUPPORTED upon receipt of this command.

The following procedure SHALL be executed upon receipt of this command.

1) If this request is broadcast, the message shall be dropped and no further processing SHALL be done.
2) The device SHALL obtain the *mibJoiningIeeeList* and *mibJoiningPolicy* from one of its currently enabled MAC Interfaces.
    a) Examine the *nwkMacInterfaceTable* and obtain an entry where State is set to ENABLED.
    b) Execute an MLME-GET.request for *mibJoiningIeeeList* and *mibJoiningPolicy*.
3) If the mibIeeeJoiningList is empty, then a Mgmt_NWK_IEEE_Joining_List_rsp SHALL be generated as follows.
    a) Status SHALL be set to SUCCESS.
    b) JoiningPolicy SHALL be set to the value of the *mibJoiningPolicy*.
    c) IeeeJoiningListTotal SHALL be set to 0.
    d) Unicast the response back to the sender of the Mgmt_NWK_IEEE_Joining_List_req.
    e) No further processing SHALL be done.
4) The device SHALL examine the StartIndex field and determine if it is less than the length of the mibJoiningIeeeList. If it is not, it SHALL do the following:
    a) A Mgmt_NWK_IEEE_Joining_List_rsp SHALL be generated with a Status value of INVALID_INDEX. No other fields shall be appended.
    b) Unicast the response back to the sender of the Mgmt_NWK_IEEE_Joining_List_req.
    c) No further processing SHALL be done.
5) The device SHALL generate a Mgmt_NWK_IEEE_Joining_List_rsp.

a) Set the Status value to SUCCESS.

b) Set the JoiningPolicy in the response to the previously obtain value of mibJoiningPolicy.

c) Set the StartIndex of the response packet equal to the value of the StartIndex in the request packet.

d) Copy complete IEEE addresses from the mibJoiningIeeeList to the IeeeJoiningList, from the Start Index, filling the payload of the packet up to the MTU.

e) Set the IeeeJoiningListTotal to the number of complete entries that were copied.

f) Unicast the response back to the sender of the Mgmt_NWK_IEEE_Joining_List_req.

2.4.3.3.12 Mgmt_NWK_Beacon_Survey_req

This command can be used by a remote device to survey the end devices to determine how many potential parents they have access to. The Mgmt_NWK_Beacon_Survey_req command (cluster ID 0x003c) SHALL be formatted as described in Figure 2-40.

Octets: Varies
TLVs

Figure 2-40. Format of the Mgmt_NWK_Beacon_Survey_req

Table 2-67 describes the fields of the Mgmt_NWK_Beacon_Survey_req command.

Table 2-67. Fields of the Mgmt_NWK_Beacon_Survey_req

Name	Type	Valid Range	Description
TLVs	TLV	Varies	The following TLVs SHALL be included in the Mgmt_NWK_Beacon_Survey_req: • Beacon Survey Configuration TLV

2.4.3.3.12.1 Beacon Survey Configuration TLV

The Beacon Survey Configuration TLV (ID=0) is variable in length and contains information about the channels and scan configuration used when performing a beacon survey. The format is listed in Figure 2-41.

Octets: Variable	1
ScanChannelListStructure	ConfigurationBitmask

Figure 2-41. Format of the Beacon Survey Configuration TLV

2.4.3.3.12.1.1 ScanChannelsListStructure

Name	Type	Valid Range	Description
ScanChan-nelsListStructure	Channel-ListStruc-ture	Variable	The list of channels and pages over which the scan is to be done. For more information on the Channel List structure see section 3.2.2.2.1.

2.4.3.3.12.1.2 Configuration Bitmask

This field indicates parameters of the Mgmt_NWK_Beacon_Survey_req. The Configuration bitmask enumerated values are specified in Table 2-68.

Table 2-68. Configuration Bitmask Values

Bit	Name	Description
0	Active or Enhanced Scan	This bit determines whether to do an Active Scan or Enhanced Active Scan. When the bit is set to 1 it indicates an Enhanced Active Scan. And in case of Enhanced Active scan EBR shall be sent with EPID filter instead of PJOIN filter.
1 – 7	Reserved	-

2.4.3.3.12.2 When Generated

This is generated by a remote device that wants to learn how many potential parents a Zigbee End Device has. The message SHALL be sent as a unicast to a single target device.

2.4.3.3.12.3 Effect on Receipt

The processing of the Mgmt_NWK_Beacon_Survey_req SHALL be done as follows:

1) If the command was broadcast it SHALL be dropped and no further processing SHALL be done.
2) If the command does not contain the mandatory TLVs listed in Figure 2-40. Format of the Mgmt_NWK_Beacon_Survey_req

Table 2-67 describes the fields of the Mgmt_NWK_Beacon_Survey_req command.

3) Table 2-67 then a Mgmt_Beacon_Survey_rsp SHALL be generated with a status of MISSING_TLV and no further processing SHALL be done.
4) If the command is received by a coordinator, the coordinator SHALL reject the command. The coordinator does not perform rejoins and thus does not need to be surveyed in this manner.
    a) The coordinator shall construct a Mgmt_NWK_Beacon_Survey_rsp with a status field value of NOT_PERMITTED and no further payload fields. It SHALL unicast the response back to the sender and no further processing SHALL be done.
5) Construct a Beacon Survey Results TLV with all sub-fields set to 0.
6) Construct a Potential Parent TLV.
    a) If the device is an End Device, set the Current parent value to the Short Address of its parent.
    b) If the device is a Router, set the current parent to 0xFFFF.
7) If the Configuration field in the Beacon Survey Configuration TLV indicates Enhanced Active Scan and the local device does not support ENHANCED_ACTIVE, then a Mgmt_Beacon_Survey_rsp SHALL be generated with a status of INV_REQUESTTYPE and no further processing SHALL be done.
8) Execute an MLME-SCAN.request with the following parameters:
    a) If the Configuration field in the Beacon Survey Configuration TLV indicates Enhanced Active Scan, set the ScanType to ENHANCED_ACTIVE. Otherwise set to ACTIVE.
    b) ScanChannels set to the list of channels contained in the Beacon Survey Configuration TLV.
9) Upon receipt of the MLME-BEACON-NOTIFY.indication process the beacons as follows:
    a) Increment the Total Beacons Field by 1.
    b) For each beacon that has a Zigbee Beacon Payload and the Extended PAN ID field of that beacon payload is equal to the nwkExtendedPanId, do the following:
        i) Increment the On-Network Beacons field.
        ii) If the End Device Capacity of the Zigbee Beacon Payload is TRUE, increment the Potential Parent Beacons field by 1.
    c) If there is no Zigbee Beacon Payload or the Extended PAN ID does not match the nwkExtendedPanId, do the following:
        i) Increment the Other Network Beacons field by 1.

d) Evaluate the beacon, potentially adding it to the Discovery Table (nwkDiscoveryTable).
e) If any of the above values reach 255, they SHALL NOT wrap and be set to 255.

10) Add up to 5 devices into the Potential Parent TLV from the contents of the nwkDiscoveryTable. Update the Count of Potential Parents accordingly.
11) Generate a ZDO Mgmt_NWK_Beacon_Survey_rsp to the sender of the request with the following TLVs
    a) Beacon Survey Results TLV.
    b) Potential Parents TLV
    c) Pan ID Conflict Report Global TLV
        i) If the device is an End Device and does not support this NIB value, this TLV may be omitted.
        ii) Note: The nwkPanIdConflictCount value in the NIB SHALL NOT be reset to 0.
12) Discard the results stored in the *nwkDiscoveryTable*.

2.4.3.4 Security Client Services

Security Client Services allow devices to configure security policies, retrieve security policies, negotiate keys, and update security tokens. Table 2-69 lists the commands supported by the Device Profile related to Security Client services.

Table 2-69. Security Client Services Commands

Security Client Services	Cluster ID	Client Transmission	Server Processing	Restricted Command
Security_Start_Key_Negotiation_req	0x0040	O	O	No
Security_Retrieve_Authentication_Token_req	0x0041	O	O	No
Security_Get_Authentication_Level_req	0x0042	O	O	No
Security_Set_Configuration_req	0x0043	O	M	No
Security_Get_Configuration_req	0x0044	O	M	No
Security_Start_Key_Update_req	0x0045	O	M	No
Security_Decommission_req	0x0046	O	M	No
Security_Challenge_req	0x0047	M	M	No

2.4.3.4.1 Security_Start_Key_Negotiation_req

The Security_Start_Key_Negotiation_req command (0x0040) shall be formatted as illustrated in Figure 2-42. This command SHALL NOT be APS encrypted regardless of whether sent before or after the device joins the network.

This command SHALL be network encrypted if the device has a network key, i.e. it has joined the network earlier and wants to negotiate or renegotiate a new link key; otherwise, if it is used prior to joining the network, it SHALL NOT be network encrypted.

Octets: Variable
TLVs

Figure 2-42. Format of the Security_Start_Key_Negotiation_req Command

Table 2-70 describes the fields of the Security_Start_Key_Negotiation_req command.

3809 **Table 2-70. Fields of the Security_Start_Key_Negotiation_req Command**

Name	Type	Valid Range	Description
TLVs	TLVs	Varies	A list of one or more TLVs. The following TLVs have specified behavior in this release of the specification: • Curve25519 Public Point TLV Other TLVs may be included.

3810 2.4.3.4.1.1 **Local TLVs**

3811 2.4.3.4.1.2 **Curve25519 Public Point TLV (ID=0)**

3812 Figure 2-43 indicates the format of the Curve25519 Public Point TLV.

Octets: 8	32
Device EUI64	Public Point

3813 **Figure 2-43. Format of the Curve25519 Public Point TLV**

3814 Table 2-71 describes the fields of the Curve25519 Public Point TLV.

3815 **Table 2-71. Fields of the Curve25519 Public Point TLV**

Field	Description
Device EUI64	This indicates the EUI64 of the device that generated the public point.
Public Point	The 32-byte Curve public point.

3816 2.4.3.4.1.3 **When Generated**

3817 The Security_Start_Key_Negotiation_req is generated from a local device that wants to start negotiation of an en-
3818 cryption key. Typically, this is used to negotiate a Trust Center Link Key during the joining process prior to becom-
3819 ing fully authorized on the network. However, it can be used after joining a network as well. Refer to section 4.6.3.5.

3820 The security primitives for key negotiation are the APSME-KEY-NEGOTIATION primitives and are used by the
3821 stack to manage the process. See section 4.4.9 for more details. Their interaction with the over-the-air messages can
3822 be found in Figure 4-6.

3823 When negotiating a Trust Center Link Key the device SHALL send at least the following TLV:

3824 • Curve25519 Public Point TLV

3825 It is EXPECTED that the sending device has already been told the selected Key Negotiation Protocol and selected
3826 Pre-Shared Secrets of the target device prior to sending this message. The sending device can learn the Supported
3827 Key Negotiation Methods in one of two possible ways: (1) in case of on-network key negotiation, the device sends
3828 first a Node Descriptor Request advertising its own supported key negotiation methods and the Node Descriptor Re-
3829 sponse will contain the selected Key Negotiation Protocol and selected Pre-Shared secret; (2) in case of off-network
3830 key negotiation, the Trust Center sends a Security Start Key Update Request with the selected Key Negotiation Pro-
3831 tocol and selected Pre-Shared secret, after it has received the TLVs conveyed in a Network Commissioning request.
3832 If the sending device supports multiple mechanisms, via implementation-specific configuration it SHALL choose
3833 one that is supported by the target device.

2.4.3.4.1.4 Effect on receipt

The Device EUI64 within the Curve25519 Public Point TLV SHALL represent the EUI64 of the device that is requesting the key negotiation with the receiving device. The processing of the message SHALL be done as follows:

1. Execute the General TLV Processing Rules in Annex I
   a. If the outcome is to reject the message, do the following.
      i. If the message was broadcast, no response is generated.
      ii. If the message is unicast, a Security_Key_Negotiation_rsp SHALL be generated with a status as returned by the General TLV Processing rules Key Exchange. The response SHALL be sent back to the sender of the Security_Retrieve_Authentication_Token_req.
      iii. No further processing SHALL be done.
   b. Otherwise, continue processing.
2. If the Curve25519 Public Point TLV is not present, then a ZDO Security_Key_Negotiation_rsp SHALL be generated with a status of MISSING_TLV.
3. Generate an APSME-KEY-NEGOTIATION.indication with the following parameters:
   a. The RequestedKeyNegotiationMethod SHALL be set to the value conveyed in the Node_Desc_rsp Selected Key Negotiation Method TLV or Security_Start_Key_Update_req Selected Key Negotiation Method TLV.
   b. The PartnerLongAddress SHALL be set to the Device EUI64 within the Curve25519 Public Point TLV.
   c. The PublicPointData SHALL be set to the public point from the Curve25519 Public Point TLV.
   d. If the ZDO frame was contained within an APS Command Relay Message Downstream, then it SHALL do the following
      i. Set RelayCommand to TRUE
      ii. Set RelayLongAddress to the address of the Device that sent the Network Data frame.

2.4.3.4.2 Security_Retrieve_Authentication_Token_req

The Security_Retrieve_Authentication_Token_req command (0x0041) shall be formatted as illustrated in Figure 2-44. This command SHALL be APS encrypted.

Octets: Variable
TLVs

Figure 2-44. Format of the Security_Retrieve_Authentication_Token_req Command

Table 2-72 describes the fields of the Security_Start_Key_Negotiation_req command.

Table 2-72. Fields of the Security_Retrieve_Authentication_Token_req Command

Name	Type	Valid Range	Description
TLVs	TLVs	Varies	A list of one or more TLVs. The following TLVs have specified behavior in this release of the specification: • Authentication Token ID TLV Other TLVs may be included.

This command is used to retrieve a security token that can be used for future authentication exchanges. Security tokens could take multiple forms such as certificates, public keys, or symmetric passphrase. As of this Revision of this specification, only a symmetric passphrase is supported. The current use of this command is to obtain a new passphrase token. The passphrase token is intended to be good for the life of the device on that network. Previously, the device SHALL have been added to the keytable of the Trust Center during the APS update device. Once the device has obtained a new passphrase, replacing either a well-known pre-shared secret or one derived from an install code

or passcode, it is locked down and not allowed to be replaced automatically. A Trust Center MAY administratively reset the device's security and thus allow it to join again and get a new token.

The passphrase used to join the network is intended to be used only once and then the device SHALL update it. The initial passphrase is either well-known (unauthenticated) or is the install code derived link key (authenticated). Once passphrase is updated it is never intended to be changed again for the life of the device on the network. The key negotiation leverages the passphrase and the devices need to avoid a circumstance where there is a passphrase mismatch, which could prevent the devices from ever successfully negotiating a symmetric link key again.

2.4.3.4.2.1 Local TLVs

2.4.3.4.2.2 Authentication Token ID TLV (ID=0)

The Authentication Token ID TLV is formatted as shown in Figure 2-45.

Octets: 1
TLV Type Tag ID

Figure 2-45. Authentication Token ID TLV

Table 2-73 describes the fields of the Authentication Token ID TLV.

Table 2-73. Requested Token ID TLV

Field	Description
TLV Type Tag ID	The Global TLV Type Tag ID being requested for an authentication token.

2.4.3.4.2.3 When Generated

This command is used to request a unique device specific authentication token that can be used for future key renegotiation. This token can be used across a replacement of the Trust Center.

A device SHALL include the authentication token type that it supports by sending the Authentication Token ID TLV with the Global TLV Type Tag ID. The only supported authentication token in this specification is 128-bit Symmetric Passphrase Global TLV.

By sending the TLV Type Tag ID this potentially allows a future specification to use alternate tokens. For example, the Type Tag ID requested could be an operational certificate and the Trust Center could sign the ephemeral public key the joiner used during joining and then send it back to the device.

Authentication tokens are only updated with this command by a device requesting one from the Trust Center. This is not used for Partner Link Key Negotiation.

2.4.3.4.2.4 Effect on receipt

Upon receipt, a device that is not the Trust Center SHALL respond with a Security_Retrieve_Authentication_Token_rsp with a status of NOT_SUPPORTED and no further processing SHALL be done. If the received message is not APS encrypted, or it is a broadcast, then the message SHALL be dropped and no further processing SHALL be done.

Obtaining a security token of a specific type SHALL only be done once during join. The token is intended to be good for the life of the device on that network. In this Revision of the specification, only the 128-bit Symmetric Passphrase is a valid token type, but to allow for future security extensions, obtaining a security token of a different type may be permitted, based on the Trust Center policy. Previously, the device SHALL have been added to the key-table during the APS update device. Once the device has obtained a new passphrase, replacing either a well-known pre-shared secret or one derived from an install code, it is locked down and not allowed to be replaced automatically. A Trust Center MAY administratively reset the device's security and thus allow it to join again and get a new token.

The Trust Center SHALL perform the following:

1. Execute the General TLV Processing Rules in Annex I.
   a. If the outcome is to reject the message, do the following:
      i. If the message was broadcast, no response is generated.
      ii. If the message is unicast, a Security_Retrieve_Authentication_Token_rsp SHALL be generated with a status of INVALID_TLV. The response SHALL be sent back to the sender of the Security_Retrieve_Authentication_Token_req.
      iii. No further processing SHALL be done.
   b. Otherwise, continue processing.
2. The Trust Center SHALL search *apsDeviceKeyPairSet* table in the AIB for an entry that matches the EUI64 of the request.
   a. If none is found then a Security_Retrieve_Authentication_Token_rsp SHALL be generated to the requesting device with a status of NOT_PERMITTED, and no further processing SHALL be done.
   b. Otherwise, continue processing.
3. If the Authentication Token ID TLV is not present then the following steps SHALL be done.
   a. A Security_Retrieve_Authentication_Token_rsp SHALL be generated to the requesting device with a status of INVALID_TLV, and no further processing SHALL be done.
4. The Trust Center SHALL examine the TLV Tag ID in the Authentication Token ID TLV received in the message.
   a. If the TLV Tag ID is not 69, 128-bit Symmetric Passphrase Global TLV then a Security_Retrieve_Authentication_Token_rsp SHALL be generated to the requesting device with a status of INV_REQUESTTYPE, and no further processing SHALL be done.
5. The Trust Center SHALL examine the value of PassphraseUpdateAllowed for the entry of the apsDeviceKeyPairSet.
   a. If this value is set to FALSE then a Security_Retrieve_Authentication_Token_rsp SHALL be generated to the requesting device with a status of NOT_PERMITTED, and no further processing SHALL be done.
   b. Otherwise, continue processing.
6. The Trust Center SHALL generate a random 128-bit number with a cryptographically secure random number generator.
7. The Trust Center SHALL store the value as the Passphrase value for the associated entry of the *apsDeviceKeyPair* table AIB value.
8. The Trust Center SHALL construct a 128-bit Symmetric Passphrase Global TLV containing the value.
9. The Trust Center SHALL generate a Security_Retrieve_Authentication_Token_rsp to the sender of the request with a status of SUCCESS and the created TLV.
10. The Trust Center SHALL set the PassphraseUpdateAllowed value to FALSE for the associated entry of the *apsDeviceKeyPair* table AIB value.

2.4.3.4.3 Security_Get_Authentication_Level_req

This command allows a device to query the trust center about a 3rd party device to determine how it is authenticated on the network. This enables the querying device to determine if that 3rd party has the minimum required authentication level for application communication.

The Security_Get_Authentication_Level_req command (ClusterID=0x0042) shall be formatted as illustrated in Figure 2-46. It SHALL have APS encryption.

Octets: Variable
TLVs

Figure 2-46. Format of the Security_Get_Authentication_Level_req Command

Table 2-74 describes the fields of the Security_Get_Authentication_Level_req command.

Table 2-74 Fields of the Security_Get_Authentication_Level_req Command

Name	Type	Valid Range	Description
TLVs	TLVs	Varies	A list of one or more TLVs. The following TLVs have specified behavior in this release of the specification: • Target IEEE Address TLV Other TLVs may be included.

2.4.3.4.3.1 Local TLVs

2.4.3.4.3.2 Target IEEE Address TLV (ID=0)

The format of the Target IEEE Address TLV is shown in Figure 2-47.

Octets: 8
IEEEAddrOfInterest

Figure 2-47. Format of the Target IEEE Address TLV

Table 2-75 specifies the fields of the Target IEEE Address TLV.

Table 2-75. Fields of the Target IEEE Address TLV

Name	Type	Valid Range	Description
IEEEAddrOfInterest	64-bit IEEE address	Any	Extended address of the device whose security level is requested.

2.4.3.4.3.3 When Generated

The Security_Get_Authentication_Level_req is generated by the local device wishing to find out the authentication level of another device on the network. The command SHALL be unicast to the trust center. This command SHALL be APS encrypted.

2.4.3.4.3.4 Effect on Receipt

The following SHALL occur on the receipt of the Security_Get_Authentication_Level_req.

1. If the Security_Get_Authentication_Level_req command is broadcast it SHALL be dropped and no further processing SHALL be done.
2. If the Security_Get_Authentication_Level_req is not APS encrypted it SHALL be dropped and no further processing SHALL be done.
3. If the receiving device is NOT the Trust Center then a Security_Get_Authentication_Level_rsp SHALL be generated with a status of NOT_AUTHORIZED and no further processing SHALL be done.
4. Execute the General TLV Processing Rules in Annex I.4.7.
5. If the Target IEEE Address TLV is not present in the message the receiver SHALL generate a Security_Authentication_Level_rsp with a status of INV_REQUESTTYPE and no further processing SHALL be done.

6. The receiving device SHALL examine the *apsDeviceKeyPairSet* table of the AIB to find the entry matching the IEEEAddrOfInterest present in the Target IEEE Address TLV.
    a. If no matching entry is found then a Security_Get_Authentication_Level_rsp SHALL be generated with a status of NO_MATCH and no further processing SHALL be done.
7. If the IEEEAddrRemoteNode is the address of the Trust Center or 0xFFFFFFFFFFFFFFFF, the receiver SHALL generate a Security_Authentication_Level_rsp with a status of INV_REQUESTTYPE and no further processing SHALL be done.
8. The Device SHALL create a Security_Get_Authentication_Leve1_rsp with the following values:
    a. Set the status of response to SUCCESS
    b. Create a Device Authentication Level TLV
        a. Set the IEEEAddrRemoteNode of the response to the IEEEAddrOfInterest received in the request frame.
        b. From the matching entry in the apsDeviceKeyPairSet table set the InitialJoinMethod value in the Device Authentication Level TLV to the value of the InitialJoinAuthentication value from the AIB entry.
        c. From the matching entry in the apsDeviceKeyPairSet table set the ActiveLinkKeyType value in the Device Authentication Level TLV to the value of the PostJoinKeyUpdateMethod value from the AIB entry.

2.4.3.4.4 Security_Set_Configuration_req

The Security_Set_Configuration_req allows the Trust Center to change configuration options for a particular device.

The format of the message is in Figure 2-48. This command SHALL be APS encrypted when operating in a centralized security network. When operating in a distributed security network the command MAY be APS encrypted.

Octets: Varies
TLVs

Figure 2-48. Format of the Security_Set_Configuration_req Command

Table 2-76 specifies the fields of the Security_Set_Configuration_req command.

Table 2-76. Fields of the Security_Set_Configuration_req Command

Name	Type	Range	Description
TLVs	TLVs	Varies	A list of one or more TLVs. The following TLVs have specified behavior in this Revision of the specification: • Next PAN ID Global TLV • Next Channel Change Global TLV • Configuration Parameters Global TLV Other TLVs may be included.

The fields of the command Security_Set_Configuration_req are specified in Table 2-76. The following TLVs MAY be present:

- Next PAN ID Change Global TLV
- Next Channel Change Global TLV
- Configuration Parameters Global TLV

2.4.3.4.4.1 Local TLVs

There are no Local TLVs defined for this command.

2.4.3.4.4.2 When Generated

This is generated by the Trust Center when it wants to change configuration settings of the device. In distributed security networks, it MAY be generated by any device that wants to change the configuration settings of a remote device. In a distributed security network, it is permissible to send this command as a broadcast.

2.4.3.4.4.3 Effect on Receipt

When operating in a centralized security network, on receipt of a Security_Set_Configuration_req sent to the broadcast address, the device SHALL drop the message and no further processing SHALL be done.

When operating in a centralized security network, on receipt of a Security_Set_Configuration_req from a device that is not the Trust Center, the receiving device SHALL generate a Security_Set_Configuration_rsp with a status of NOT_AUTHORIZED. No further processing SHALL be done.

When operating in a distributed network, this command MAY be broadcast or unicast and MAY or MAY NOT be APS encrypted. The command is accepted in all those cases.

On receipt of a Security_Set_Configuration_req by the Trust Center, the Trust Center device SHALL generate a Security_Set_Configuration_rsp with a status of NOT_AUTHORIZED. No further processing SHALL be done.

Processing of the message SHALL be done as follows:

1. Execute the General TLV Processing rules in Annex I.
2. If the result of the processing indicates a failure, then do the following.
    a. If the command was unicast, the receiver SHALL transmit a Security_Set_Configuration_rsp to the sender with the status code as returned from the General TLV Processing rules.
    b. If the command was broadcast, no response is generated.
    c. For all cases when the TLV processing fails, no further processing SHALL be done.
3. Process the TLVs in the message as follows:
    a. Upon receipt of the Configuration Parameters Global TLV, the stack SHALL modify the value of the corresponding information base value as referenced in Table 4 36.
    b. Upon receipt of the Next PAN ID Global TLV, the stack SHALL modify the NIB value of the nwkNextPanId according to the value received in the TLV. Setting the nwkNextPanId to the broadcast PAN ID is allowed. It indicates that any PAN ID MAY be used as the next PAN ID.
    c. Upon receipt of the Next Channel Change Global TLV the stack SHALL modify the NIB value of the nwkNextChannelChange in the NIB according to the value received in the TLV if it matches one of the Supported Channels of an interface in the nwkMacInterfaceTable.

4. If no TLVs were processed in step 3, do the following:
    a. If the command was broadcast, no more processing SHALL take place.
    b. If the command was unicast, send a Security_Set_Configuration_rsp with a status MISSING_TLV to the sender of the request.

Note that an Overall Status of NOT_SUPPORTED for the Security_Set_Configuration_rsp is reserved for stacks prior to Revision 23 that do not understand the Security_Set_Configuration_req command at all.

2.4.3.4.5 Security_Get_Configuration_req

This command is used by a device to retrieve a remote device's security configuration. The Security_Get_Configuration_req command (cluster ID = 0x0044) is formatted as illustrated in Figure 2-49.

This command SHALL be APS encrypted in centralized security mode. It MAY be APS encrypted in distributed security mode.

Octets: 1	1	…
TLV Count	TLV ID	…

Figure 2-49. Format of the Security_Get_Configuration_req Command Frame

Table 2-77 specifies the fields of the Security_Get_Configuration_req command frame.

Table 2-77. Fields of the Security_Get_Configuration_req Command Frame

Name	Type	Valid Range	Description
TLV Count	Integer	0 – 255	The number of TLV IDs contained in the message. Note that the maximum value for this count will be dependent on the underlying maximum size of the message as allowed by fragmentation.
TLV ID	Integer	0 – 255	The ID of each TLV that is being requested.

2.4.3.4.5.1 When Generated

This is generated by a device that wants to retrieve the configuration of a remote device.

2.4.3.4.5.2 Effect on Receipt

If the command was broadcast it SHALL be rejected and silently dropped.

In a centralized security network, if the local device receives this command from a remote device that is not the Trust Center the command SHALL be rejected. The receiver SHALL generate a Security_Get_Configuration_rsp with a status of NOT_AUTHORIZED. No further processing SHALL be done.

The following processing SHALL be done.

1. Construct a Security_Get_Configuration_rsp command with a status of SUCCESS.
2. For each TLV ID listed in the message, the device SHALL determine if the TLV is known to the local device and has a value.
    a. If the TLV is unknown or the local device has no value for that TLV, it SHALL be skipped and processing will continue with the next TLV. For example, if the device has no Curve25519 Public Point then it would ignore a request for its Curve25519 Public Point TLV.
3. If the TLV ID is equal to the ID of PAN ID Conflict Report Global TLV, then the following SHALL occur.
    a. Construct a PAN ID Conflict Report Global TLV using the current NIB value of nwkPanIdConflictCount.
    b. Set the NIB value nwkPanIdConflictCount to 0.
4. The corresponding TLV SHALL be constructed and appended to the ZDO message.
5. If appending the TLV exceeds the MTU for the message then the following SHALL be done.
    a. Abort processing. Construct and send a Security_Get_Configuration_rsp with a STATUS of FRAME_TOO_LARGE and no other payload.
6. Transmit the Security_Get_Configuration_rsp to the sender of the request.

2.4.3.4.6 Security_Start_Key_Update_req

This command is used by the Trust Center to trigger the receiving device to start its supported link key update mechanism. The Security_Start_Key_Update_req SHALL NOT be APS encrypted or NWK encrypted if the link key update mechanism is done as part of the initial join and before the receiving device has been issued a network key.

The Security_Start_Key_Update_req SHALL be both APS encrypted and NWK encrypted if the link key update mechanism is performed to refresh the link key when the receiving device has the network key and has previously successfully joined the network. The Security_Start_Key_Update_req command (cluster ID = 0x0045) is formatted as illustrated in Figure 2-50.

Octets: Varies
TLVs

Figure 2-50. Format of the Security_Start_Key_Update_req

Table 2-78 specifies the fields of the Security_Start_Key_Update_req command.

Table 2-78. Fields of the Security_Start_Key_Update_req

Name	Type	Valid Range	Description
TLVs	TLV	Varies	The Security_Start_Key_Update_req SHALL include the following TLVs: • Selected Key Negotiation Method TLV • Fragmentation Parameters Global TLV Other TLVs may be included.

2.4.3.4.6.1 Local TLVs

2.4.3.4.6.2 Selected Key Negotiation Method (ID=0)

This indicates the key negotiated method that the sending device would like to negotiate with the receiver. The format is defined in Figure 2-51.

1	Octets: 1	Octets: 8
Selected Key Negotiation Protocol Enumeration	Selected Pre-shared Secret Enumeration	Sending Device EUI64

Figure 2-51. Selected Key Negotiation Method TLV

Table 2-79 indicates the fields of the Selected Key Negotiation Method TLV.

Table 2-79. Fields of the Selected Key Negotiation Method TLV

Name	Type	Valid Range	Description
Selected Key Negotiation Protocol Enumeration	Enum	0 – 2	The enumeration of the key negotiation method the sender is requesting to use in key negotiation.
Selected Pre-shared Secret Enumeration	Enum	0 – 4	The enumeration indicating the pre-shared secret that the sending device is requesting to be used in the key negotiation.
Sending Device EUI64	EUI64	Any	The value of the EUI64 of the device sending the message. This field SHALL always be present.

Table 2-80 defines the Selected Key Negotiation Protocol Enumeration.

Table 2-80. Selected Key Negotiation Protocol Enumeration

Enumerated Value	Description
0	Reserved (Zigbee 3.0 Mechanism)
1	SPEKE using Curve25519 with Hash AES-MMO-128
2	SPEKE using Curve25519 with Hash SHA-256
3 – 255	Reserved

Table 2-81 defines the Selected Pre-shared Secret Enumeration.

Table 2-81. Selected Pre-shared Secret Enumeration

Enumerated Value	Description
0	Symmetric Authentication Token
1	Pre-configured link-ley derived from installation code
2	Variable-length pass code (for PAKE protocols)
3	Basic Authorization Key
4	Administrative Authorization Key
5 – 254	Reserved
255	Anonymous Well-Known Secret

2.4.3.4.6.3 When Generated

This command is generated by the Trust Center when it wants to trigger the key update process for a device.

2.4.3.4.6.4 Effect on Receipt

On receipt, this command SHALL be processed as follows:

1) If the apsTrustCenterAddress is all F's or if apsTrustCenterAddress is not all F's and the command was not sent by the Trust Center, the receiver SHALL generate a Security_Start_Key_Update_rsp with a status of NOT_AUTHORIZED.
2) If the mandatory TLVs from Table 2-78 are not included, then a Security_Start_Key_Update_rsp SHALL be generated with a status of INV_REQUESTTYPE and no further processing SHALL be done.
3) If apsTrustCenterAddress is unset, the receive SHALL set it with the value of the Sending Device EUI64 field of the Selected Key Negotiation Method TLV.
4) Examine the Selected Key Negotiation Method TLV and determine if the device supports the selected key negotiation methods. If it does not, then a Security_Start_Key_Update_rsp SHALL be generated with a status of NO_MATCH. No further processing SHALL be done.
5) The stack MAY notify the higher layer by passing the contents of the Selected Key Negotiation Method TLV. The stack is responsible for kicking off Key Negotiation or static link key update using one of the locally supported methods.
6) Generate a ZDO Security_Start_Key_Update_rsp with a status of SUCCESS.

2.4.3.4.7 Security_Decommission_req

This command is sent by the Trust Center to inform of the decommissioning of a 3rd party device on the network. The receiving device can use this to clear out any security keys and bindings associated with that 3rd party device. This

message SHALL be sent unicast with APS encryption for a centralized network and no APS encryption for a distributed network.

The Security_Decommission_req (Cluster ID=0x0046) is formatted as illustrated in Figure 2-52.

Octets: Varies
TLVs

Figure 2-52. Format of the ZDO Security_Decommission_req Command

Table 2-82 indicates the fields of the ZDO Security_Decommission_req command.

Table 2-82. Fields of the ZDO Security_Decommission_req Command

Name	Type	Valid Range	Description
TLVs	TLVs	Varies	A list of one or more TLVs. The following TLVs have specified behavior in this Revision of the specification: • Device EUI64 List TLV Other TLVs may be included.

2.4.3.4.7.1 Local TLVs

The Local TLVs for the Security_Decommission_req command frame are as follows.

2.4.3.4.7.2 Device EUI64 List TLV (ID=0)

The format of the Device EUI64 List TLV SHALL be as formatted in Figure 2-53.

Octets: 1	8	…
EUI64 Count	EUI64	…

Figure 2-53. Format of the Device EUI64 List TLV

Table 2-83 indicates the fields of the Device EUI64 List TLV.

Table 2-83. Fields of the Device EUI64 List TLV

Name	Type	Valid Range	Description
EUI64 Count	Integer	0x00 – 0xFF	The number of EUI64 fields within the TLV. Note that the maximum value for this count will be dependent on the underlying maximum size of the message as allowed by fragmentation.
EUI64	EUI64	0x0000000000000000 – 0xFFFFFFFFFFFFFFFF	An EUI64 that shall trigger decommissioning operations.

2.4.3.4.7.3 When Generated

This command is generated when the Trust Center has administratively removed a device from the list of authorized devices and wishes to inform other devices about that action. It is NOT used to actually remove that device.

2.4.3.4.7.4 Effect on Receipt

On receipt the following processing SHALL take place.

1) If the command is broadcast it SHALL be silently dropped. No further processing SHALL be done.
2) If the command is unicast on a centralized network with no APS encryption, a ZDO Security_Decommission_rsp SHALL be generated with a status code of NOT_AUTHORIZED. No further processing SHALL be done.
3) If the receiving device is the Trust Center the command SHALL be rejected.
    a) A ZDO Security_Decommission_rsp SHALL be generated with a status code of NOT_AUTHORIZED. No further processing SHALL be done.
4) Execute the General TLV Processing Rules in Annex I.4.7.
5) If the command does not have at least one Device EUI64 List TLV present in the message, it SHALL be rejected.
    a) The receiver SHALL generate a ZDO Security_Decommission_rsp with a status of INV_REQUESTTYPE. No further processing SHALL be done.
6) The receiving device SHALL compare its local EUI64 to all EUI64 in the Security Decommission Req EUI64 TLV. If any EUI64 matches the device's local EUI64 it SHALL be rejected.
    a) The device SHALL generate a ZDO Security_Decommission_rsp with a status of INV_REQUESTTYPE.
7) The receiving device SHALL compare the value of all EUI64 values the Security Decommission Req EUI64 TLV to the DeviceAddress element of all entries in the *apsDeviceKeyPairSet* of the AIB.
    a) If any entry matches it SHALL be deleted.
8) The receiving device SHALL compare the value of all EUI64 in the Security Decommission Req EUI64 TLV to the EUI64 of each binding table entry.
    a) If any entry matches it SHALL be deleted by issuing an APSME-UNBIND.request.
9) Note that the use of the wildcard EUI64 address of 0xFFFFFFFFFFFFFFFF is not allowed and SHALL be ignored.
10) The ZDO MAY inform the NLME of each decommissioned EUI64 allowing the NLME layer to clean up any network layer data related to that device.
11) The device SHALL issue an APS encrypted ZDO Security_Decommission_rsp with the following fields
    a) The Status SHALL be set to SUCCESS if at least one EUI64 matched and resulted in the device making changes to its internal tables.
    b) Otherwise the Status SHALL be set to NOT_FOUND.

2.4.3.4.8 Security_Challenge_req

This command is used by a device to verify the latest frame counter value of another device. The Security_Challenge_req (Cluster ID = 0x0047) is formatted as illustrated in Figure 2-54.

Octets: Varies
TLVs

Figure 2-54. Format of the Security_Challenge_req

2.4.3.4.8.1 Local TLVs

Table 2-84 defines the Local scoped TLVs for this message.

Table 2-84. Global TLVs for Security_Challenge_req

Tag ID	Name
0x00	APS Frame Counter Challenge

2.4.3.4.8.2 APS Frame Counter Challenge TLV

Figure 2-55 illustrates the format of the APS Frame Counter Challenge TLV.

Octets: 8	8
Sender EUI64	Challenge Value

Figure 2-55. Format of the APS Frame Counter Challenge TLV

Table 2-85 describes the fields of the APS Frame Counter Challenge TLV.

Table 2-85. Fields of the APS Frame Counter Challenge TLV

Field	Description
Sender EUI64	The EUI64 of the device that generated the frame.
Challenge Value	A randomly generated 64-bit value sent to a device to prove they have the link key. This allows the initiator to detect replayed challenge response frames.

2.4.3.4.8.3 When Generated

This command is generated when a device wants to challenge another device to verify it has the latest cryptographic data.

This message SHALL NOT be APS encrypted.

2.4.3.4.8.4 Effect on Receipt

1. If the message was broadcast it SHALL be dropped and no further processing SHALL be done.
2. If the message did not include the APS Frame Counter Challenge TLV do the following.
    a. Generate a ZDO Security_Challenge_rsp with a status of MISSING_TLV and send to the device that generated the request.
    b. No further processing SHALL be done.
3. Search the *apsDeviceKeyPairSet* table of the AIB for any entry where the DeviceAddress matches the Sender EUI64 value of the APS Frame Counter Challenge TLV
4. If no match can be found, do the following.
    a. Generate a ZDO Security_Challenge_rsp with a status of NO_MATCH and send to the device that generated the request.
    b. No further processing SHALL be done.
5. Otherwise, follow the procedure in section 4.6.3.8.4.

2.4.4 Server Services

The Device Profile Server Services support the processing of device and service discovery requests, bind requests, unbind requests, and network management requests. Additionally, Server Services support transmission of these responses back to the requesting device.

2.4.4.1 ZDO Response Requirements

A device SHALL be required to support generation of the correct, corresponding ZDO response to all ZDO requests including ZDO messages defined in a future version of this specification. Server Processing marked optional in Table 2-86, Table 2-96, and Table 2-100 allow for the server to use NOT_SUPPORTED as the status code in the response to indicate the lack of support. ZDO requests unknown to the device SHALL be treated as unsupported and also use a NOT_SUPPORTED status code to indicate the device's lack of support for that feature. See below for construction of ZDO responses to unsupported requests. For all broadcast addressed requests (of any broadcast address type) to the server, if the command is not supported, the server SHALL drop the packet. No error status SHALL be unicast back to the Local Device for any broadcast addressed client request including, but not limited to, requests which are not supported on the server.

For all unicast addressed requests to the server, if the command is not supported, the server SHALL formulate a response packet including the response Cluster ID and status fields only. The response Cluster ID SHALL be created by taking the request Cluster ID and setting the high order bit to create the response Cluster ID. The status field SHALL be set to NOT_SUPPORTED. The resulting response SHALL be unicast to the requesting client.

2.4.4.2 Device and Service Discovery Server

Table 2-86 lists the commands supported by the Device and Service Discovery Server Services device profile. Each of these commands will be discussed in the following sections. For receipt of the Device_annce command, the server SHALL check all internal references to the IEEE and 16-bit NWK addresses supplied in the request. For all references to the IEEE address in the Local Device, the corresponding NWK address supplied in the Device_annce SHALL be substituted. For any other references to the NWK address in the Local Device, the corresponding entry SHALL be marked as not having a known valid 16-bit NWK address, even if the IEEEAddr field in the message carries the value of 0xffffffffffffffff. The server SHALL NOT supply a response to the Device_annce.

Table 2-86. Device and Service Discovery Server Service Primitives

Device and Service Discovery Server Services	Cluster ID	Server Processing
NWK_addr_rsp	0x8000	M
IEEE_addr_rsp	0x8001	M
Node_Desc_rsp	0x8002	M
Power_Desc_rsp	0x8003	M
Simple_Desc_rsp	0x8004	M
Active_EP_rsp	0x8005	M
Match_Desc_rsp	0x8006	M
Complex_Desc_rsp	0x8010	Deprecated

Device and Service Discovery Server Services	Cluster ID	Server Processing
User_Desc_rsp	0x8011	Deprecated
User_Desc_conf	0x8014	Deprecated
Parent_annce_rsp	0x801f	M
System_Server_Discovery_rsp	0x8015	O
Discovery_store_rsp	0x8016	Deprecated
Node_Desc_store_rsp	0x8017	Deprecated
Power_Desc_store_rsp	0x8018	Deprecated
Active_EP_store_rsp	0x8019	Deprecated
Simple_Desc_store_rsp	0x801a	Deprecated
Remove_node_cache_rsp	0x801b	Deprecated
Find_node_cache_rsp	0x801c	Deprecated
Extended_Simple_Desc_rsp	0x801d	Deprecated
Extended_Active_EP_rsp	0x801e	Deprecated

2.4.4.2.1 NWK_addr_rsp

The NWK_addr_rsp command (ClusterID=0x8000) SHALL be formatted as illustrated in Figure 2-56.

Octets: 1	8	2	0/1	0/1	Variable
Status	IEEEAddr RemoteDev	NWKAddr RemoteDev	Num AssocDev	StartIndex	NWKAddr AssocDevList

Figure 2-56. Format of the NWK_addr_rsp Command Frame

Table 2-87 specifies the fields of the NWK_addr_rsp command frame.

Table 2-87. Fields of the NWK_addr_rsp Command Frame

Name	Type	Valid Range	Description
Status	Integer	SUCCESS, INV_REQUESTTYPE, or DEVICE_NOT_FOUND	The status of the NWK_addr_req command.

Name	Type	Valid Range	Description
IEEEAddrRemoteDev	Device Address	An extended 64-bit, IEEE address	64-bit address for the Remote Device.
NWKAddrRemoteDev	Device Address	A 16-bit, NWK address	16-bit address for the Remote Device.
NumAssocDev	Integer	0x00 – 0xff	Count of the number of 16-bit short addresses to follow. If the RequestType in the request is Extended Response and there are no associated devices on the Remote Device, this field SHALL be set to 0. If an error occurs or the Request Type in the request is for a Single Device Response, this field SHALL NOT be included in the frame.
StartIndex	Integer	0x00 – 0xff	Starting index into the list of associated devices for this report. If the RequestType in the request is Extended Response and there are no associated devices on the Remote Device, this field SHALL NOT be included in the frame. If an error occurs or the Request Type in the request is for a Single Device Response, this field SHALL NOT be included in the frame.
NWKAddrAssocDevList	Device Address List	List of NumAssocDev 16-bit short addresses, each with range 0x0000 – 0xffff	A list of 16-bit addresses, one corresponding to each associated device to Remote Device; The number of 16-bit network addresses contained in this field is specified in the NumAssocDev field. If the RequestType in the request is Extended Response and there are no associated devices on the Remote Device, this field SHALL NOT be included in the frame. If an error occurs or the Request Type in the request is for a Single Device Response, this field SHALL NOT be included in the frame.

2.4.4.2.1.1 When Generated

The NWK_addr_rsp is generated by a Remote Device in response to a NWK_addr_req command inquiring as to the NWK address of the Remote Device or the NWK address of an address held in the neighbor table (see section 2.4.3.1.1.2 for a detailed description). The destination addressing on this command is unicast.

2.4.4.2.1.2 Effect on Receipt

On receipt of the NWK_addr_rsp command, the recipient is either notified of the status of its attempt to discover a NWK address from an IEEE address or notified of an error. If the NWK_addr_rsp command is received with a Status of SUCCESS, the remaining fields of the command contain the appropriate discovery information, according to the RequestType as specified in the original NWK_Addr_req command. Otherwise, the Status field indicates the error and the NumAssocDev, StartIndex, and NWKAddrAssocDevList fields SHALL NOT be included.

2.4.4.2.2 IEEE_addr_rsp

The IEEE_addr_rsp command (ClusterID=0x8001) SHALL be formatted as illustrated in Figure 2-57.

Octets: 1	8	2	0/1	0/1	Variable
Status	IEEEAddr RemoteDev	NWKAddr RemoteDev	NumAssocDev	StartIndex	NWKAddr AssocDevList

Figure 2-57. Format of the IEEE_addr_rsp Command Frame

Table 2-88 specifies the fields of the IEEE_addr_rs command frame.

Table 2-88. Fields of the IEEE_addr_rsp Command Frame

Name	Type	Valid Range	Description
Status	Integer	SUCCESS, INV_REQUESTTYPE or DEVICE_NOT_FOUND	The status of the IEEE_addr_req command.
IEEEAddrRemoteDev	Device Address	An extended 64-bit, IEEE address	64-bit address for the Remote Device.
NWKAddrRemoteDev	Device Address	A 16-bit, NWK address	16-bit address for the Remote Device.
NumAssocDev	Integer	0x00 – 0xff	Count of the number of 16-bit short addresses to follow. If the RequestType in the request is Extended Response and there are no associated devices on the Remote Device, this field SHALL be set to 0. If an error occurs or the RequestType in the request is for a Single Device Response, this field SHALL NOT be included in the frame.

Name	Type	Valid Range	Description
StartIndex	Integer	0x00 – 0xff	Starting index into the list of associated devices for this report. If the RequestType in the request is Extended Response and there are no associated devices on the Remote Device, this field SHALL NOT be included in the frame. If an error occurs or the RequestType in the request is for a Single Device Response, this field SHALL NOT be included in the frame.
NWKAddrAssocDevList	Device Address List	List of NumAssocDev 16-bit short addresses, each with range 0x0000 – 0xffff	A list of 16-bit addresses, one corresponding to each associated device to Remote Device; The number of 16-bit network addresses contained in this field is specified in the NumAssocDev field. If the RequestType in the request is Extended Response and there are no associated devices on the Remote Device, this field SHALL NOT be included in the frame. If an error occurs or the RequestType in the request is for a Single Device Response, this field SHALL NOT be included in the frame

2.4.4.2.2.1 When Generated

The IEEE_addr_rsp is generated by a Remote Device in response to an IEEE_addr_req command inquiring as to the 64-bit IEEE address of the Remote Device or the 64-bit IEEE address of an address held in the neighbor table (see section 2.4.3.1.2.2 for a detailed description). The destination addressing on this command SHALL be unicast.

2.4.4.2.2.2 Effect on Receipt

On receipt of the IEEE_addr_rsp command, the recipient is either notified of the status of its attempt to discover an IEEE address from an NWK address or notified of an error. If the IEEE_addr_rsp command is received with a Status of SUCCESS, the remaining fields of the command contain the appropriate discovery information, according to the RequestType as specified in the original IEEE_Addr_req command. Otherwise, the Status field indicates the error and the NumAssocDev, StartIndex, and NWKAddrAssocDevList fields SHALL NOT be included.

2.4.4.2.3 Node_Desc_rsp

The Node_Desc_rsp command (ClusterID=0x8002) SHALL be formatted as illustrated in Figure 2-58.

Octets: 1	2	See section 2.3.2.3	TLVs
Status	NWKAddrOfInterest	Node Descriptor	One or more TLVs

Figure 2-58. Format of the Node_Desc_rsp Command Frame

Table 2-89 specifies the fields of the Node_Desc_rsp command frame.

Table 2-89. Fields of the Node_Desc_rsp Command Frame

Name	Type	Valid Range	Description
Status	Integer	SUCCESS, DEVICE_NOT_FOUND, INV_REQUESTTYPE, or NO_DESCRIPTOR	The status of the Node_Desc_req command.
NWKAddrOfInterest	Device Address	16-bit NWK address	NWK address for the request.
NodeDescriptor	Node Descriptor		See the Node Descriptor format in section 2.3.2.3. This field SHALL only be included in the frame if the status field is equal to SUCCESS.
TLVs	TLV List	Varies	A set of TLVs. The Fragmentation Parameters Global TLV SHALL always be included.

2.4.4.2.3.1 Local TLVs

2.4.4.2.3.2 Selected Key Negotiation Method (ID=0)

This TLV has the same format and ID as the Selected Key Negotiation Method TLV of the Security_Start_Key_Update_req.

2.4.4.2.3.3 When Generated

The Node_Desc_rsp is generated by a remote device in response to a Node_Desc_req directed to the remote device. This command SHALL be unicast to the originator of the Node_Desc_req command.

If the Node_Desc_req frame includes the Fragmentation Parameters Global TLV the receiver can cache the information in the *apsFragmentationCacheTable*. See section 2.4.4.2.3.4 for more information.

If the Node_Desc_req frame includes at least one valid TLV, the receiver SHALL set the Frame Counter Synchronization bit in the Features & Capabilities bitmap of the apsDeviceKeyPairSet entry pertaining to the sender of the Node_Desc_req command to '1', if such an entry exists.

The remote device SHALL generate the Node_Desc_rsp command using the format illustrated in Figure 2-58. The NWKAddrOfInterest field SHALL match that specified in the original Node_Desc_req command. If the NWKAddrOfInterest field matches the network address of the remote device, it SHALL set the Status field to SUCCESS and include its node descriptor (see section 2.3.2.3) in the NodeDescriptor field.

If the NWKAddrOfInterest field does not match the network address of the remote device and it is an end device, it SHALL set the Status field to INV_REQUESTTYPE, set the ActiveEPCount field to 0, and not include the ActiveEPList field. If the NWKAddrOfInterest field does not match the network address of the remote device and it is the coordinator or a router, it SHALL set the Status field to DEVICE_NOT_FOUND, set the ActiveEPCount field to 0, and not include the ActiveEPList field. If the NWKAddrOfInterest matches the network address of one of the children of the remote device, it SHALL determine whether a node descriptor for that device is available. If a node descriptor is not available for the child indicated by the NWKAddrOfInterest field, the remote device SHALL set the Status field to NO_DESCRIPTOR and not include the NodeDescriptor field. If a node descriptor is available for the

child indicated by the NWKAddrOfInterest field, the remote device SHALL set the Status field to SUCCESS and include the node descriptor (see section 2.3.2.3) of the matching child device in the NodeDescriptor field.

The device sending the Node_Desc_rsp SHALL include the following TLVs:

- Selected Key Negotiation Method TLV.
- Fragmentation Parameters Global TLV

Devices prior to Revision 23 will not include the TLV field. The receiver SHALL still accept messages without TLVs in the response message.

2.4.4.2.3.4 Effect on Receipt

On receipt of the Node_Desc_rsp command, the recipient is either notified of the node descriptor of the remote device indicated in the original Node_Desc_req command or notified of an error. If the Node_Desc_rsp command is received with a Status of SUCCESS, the NodeDescriptor field SHALL contain the requested node descriptor. Otherwise, the Status field indicates the error and the NodeDescriptor field SHALL NOT be included.

The receiver can use the Fragmentation Parameters Global TLV to cache the sender's fragmentation capabilities in the *apsFragmentationCacheTable*. The Trust Center SHALL cache the data for all devices in the network. A regular device SHALL cache fragmentation support for the Trust Center and MAY cache data for any other device in the network.

If the core stack Revision indicated in the Node_Desc_rsp is 23 or higher, the receiver SHALL set the Frame Counter Synchronization bit in the Features & Capabilities bitmap of the apsDeviceKeyPairSet entry pertaining to the sender of the Node_Desc_rsp command to '1', if such an entry exists.

2.4.4.2.4 Power_Desc_rsp

The Power_Desc_rsp command (ClusterID=0x8003) SHALL be formatted as illustrated in Figure 2-59.

Octet: 1	2	Variable
Status	NWKAddrOfInterest	Power Descriptor

Figure 2-59. Format of the Power_Desc_rsp Command Frame

Table 2-90 specifies the fields of the Power_Desc_rsp command frame.

Table 2-90. Fields of the Power_Desc_rsp Command Frame

Name	Type	Valid Range	Description
Status	Integer	SUCCESS, DEVICE_NOT_FOUND, INV_REQUESTTYPE, or NO_DESCRIPTOR	The status of the Power_Desc_req command.
NWKAddrOfInterest	Device Address	16-bit NWK address	NWK address for the request.

Name	Type	Valid Range	Description
PowerDescriptor	Power Descriptor		See the Node Power Descriptor format in section 2.3.2.4. This field SHALL only be included in the frame if the status field is equal to SUCCESS.

2.4.4.2.4.1 When Generated

The Power_Desc_rsp is generated by a remote device in response to a Power_Desc_req directed to the remote device. This command SHALL be unicast to the originator of the Power_Desc_req command.

The remote device SHALL generate the Power_Desc_rsp command using the format illustrated in . The NWKAddrOfInterest field SHALL match that specified in the original Power_Desc_req command. If the NWKAddrOfInterest field matches the network address of the remote device, it SHALL set the Status field to SUCCESS and include its power descriptor (see section 2.3.2.4) in the PowerDescriptor field.

If the NWKAddrOfInterest field does not match the network address of the remote device and it is an end device, it SHALL set the Status field to INV_REQUESTTYPE and not include the PowerDescriptor field. If the NWKAddrOfInterest field does not match the network address of the remote device and it is the coordinator or a router, it SHALL determine whether the NWKAddrOfInterest field matches the network address of one of its children. If the NWKAddrOfInterest field does not match the network address of one of the children of the remote device, it SHALL set the Status field to DEVICE_NOT_FOUND and not include the PowerDescriptor field. If the NWKAddrOfInterest matches the network address of one of the children of the remote device, it SHALL determine whether a power descriptor for that device is available. If a power descriptor is not available for the child indicated by the NWKAddrOfInterest field, the remote device SHALL set the Status field to NO_DESCRIPTOR and not include the PowerDescriptor field. If a power descriptor is available for the child indicated by the NWKAddrOfInterest field, the remote device SHALL set the Status field to SUCCESS and include the power descriptor (see section 2.3.2.4) of the matching child device in the PowerDescriptor field.

2.4.4.2.4.2 Effect on Receipt

On receipt of the Power_Desc_rsp command, the recipient is either notified of the power descriptor of the remote device indicated in the original Power_Desc_req command or notified of an error. If the Power_Desc_rsp command is received with a Status of SUCCESS, the PowerDescriptor field SHALL contain the requested power descriptor. Otherwise, the Status field indicates the error and the PowerDescriptor field SHALL NOT be included.

2.4.4.2.5 Simple_Desc_rsp

The Simple_Desc_rsp command (ClusterID=0x8004) SHALL be formatted as illustrated in Figure 2-60.

Octet: 1	2	1	Variable
Status	NWKAddrOfInterest	Length	Simple Descriptor

Figure 2-60. Format of the Simple_Desc_rsp Command Frame

Table 2-91 specifies the fields of the Simple_Desc_rsp command frame.

Table 2-91. Fields of the Simple_Desc_rsp Command Frame

Name	Type	Valid Range	Description
Status	Integer	SUCCESS, INVALID_EP, NOT_ACTIVE, DEVICE_NOT_FOUND, INV_REQUESTTYPE or NO_DESCRIPTOR	The status of the Simple_Desc_req command.
NWKAddrOfInterest	Device Address	16-bit NWK address	NWK address for the request.
Length	Integer	0x00 – 0xff	Length in bytes of the Simple Descriptor to follow.
SimpleDescriptor	Simple Descriptor		See the Simple Descriptor format in section 2.3.2.5. This field SHALL only be included in the frame if the status field is equal to SUCCESS.

2.4.4.2.5.1 When Generated

The Simple_Desc_rsp is generated by a remote device in response to a Simple_Desc_req directed to the remote device. This command SHALL be unicast to the originator of the Simple_Desc_req command.

The remote device SHALL generate the Simple_Desc_rsp command using the format illustrated in . The NWKAddrOfInterest field SHALL match that specified in the original Simple_Desc_req command. If the endpoint field specified in the original Simple_Desc_req command does not fall within the correct range specified in Table 2-91, the remote device SHALL set the Status field to INVALID_EP, set the Length field to 0 and not include the SimpleDescriptor field.

If the NWKAddrOfInterest field matches the network address of the remote device, it SHALL determine whether the endpoint field specifies the identifier of an active endpoint on the device. If the endpoint field corresponds to an active endpoint, the remote device SHALL set the Status field to SUCCESS, set the Length field to the length of the simple descriptor on that endpoint, and include the simple descriptor (see section 2.3.2.5) for that endpoint in the SimpleDescriptor field. If the endpoint field does not correspond to an active endpoint, the remote device SHALL set the Status field to NOT_ACTIVE, set the Length field to 0, and not include the SimpleDescriptor field.

If the NWKAddrOfInterest field does not match the network address of the remote device and it is an end device, it SHALL set the Status field to INV_REQUESTTYPE, set the Length field to 0, and not include the SimpleDescriptor field. If the NWKAddrOfInterest field does not match the network address of the remote device and it is the coordinator or a router, it SHALL determine whether the NWKAddrOfInterest field matches the network address of one of its children. If the NWKAddrOfInterest field does not match the network address of one of the children of the remote device, it SHALL set the Status field to DEVICE_NOT_FOUND, set the Length field to 0, and not include the SimpleDescriptor field.

If the NWKAddrOfInterest matches the network address of one of the children of the remote device, it SHALL determine whether a simple descriptor for that device and on the requested endpoint is available. If a simple descriptor is not available on the requested endpoint of the child indicated by the NWKAddrOfInterest field, the remote device SHALL set the Status field to NO_DESCRIPTOR, set the Length field to 0, and not include the SimpleDescriptor field. If a simple descriptor is available on the requested endpoint of the child indicated by the NWKAddrOfInterest field, the remote device SHALL set the Status field to SUCCESS, set the Length field to the length of the simple descriptor on that endpoint, and include the simple descriptor (see section 2.3.2.5) for that endpoint of the matching child device in the SimpleDescriptor field.

2.4.4.2.5.2 Effect on Receipt

On receipt of the Simple_Desc_rsp command, the recipient is either notified of the simple descriptor on the endpoint of the remote device indicated in the original Simple_Desc_req command or notified of an error. If the Simple_Desc_rsp command is received with a Status of SUCCESS, the SimpleDescriptor field SHALL contain the requested simple descriptor. Otherwise, the Status field indicates the error and the SimpleDescriptor field SHALL NOT be included.

2.4.4.2.6 Active_EP_rsp

The Active_EP_rsp command (ClusterID=0x8005) SHALL be formatted as illustrated in Figure 2-61.

Octet: 1	2	1	Variable
Status	NWKAddrOfInterest	ActiveEPCount	ActiveEPList

Figure 2-61. Format of the Active_EP_rsp Command Frame

Table 2-92 specifies the fields of the Active_EP_rsp command frame.

Table 2-92. Fields of the Active_EP_rsp Command Frame

Name	Type	Valid Range	Description
Status	Integer	SUCCESS, DEVICE_NOT_FOUND, INV_REQUESTTYPE, or NO_DESCRIPTOR	The status of the Active_EP_req command.
NWKAddrOfInterest	Device Address	16-bit NWK address	NWK address for the request.
ActiveEPCount	Integer	0x00 – 0xff	The count of active endpoints on the Remote Device.
ActiveEPList			List of bytes each of which represents an 8-bit endpoint.

2.4.4.2.6.1 When Generated

The Active_EP_rsp is generated by a remote device in response to an Active_EP_req directed to the remote device. This command SHALL be unicast to the originator of the Active_EP_req command.

The remote device SHALL generate the Active_EP_rsp command using the format illustrated in . The NWKAddrOfInterest field SHALL match that specified in the original Active_EP_req command. If the NWKAddrOfInterest field matches the network address of the remote device, it SHALL set the Status field to SUCCESS, set the ActiveEPCount field to the number of active endpoints on that device and include an ascending list of all the identifiers of the active endpoints on that device in the ActiveEPList field.

If the NWKAddrOfInterest field does not match the network address of the remote device and it is an end device, it SHALL set the Status field to INV_REQUESTTYPE, set the ActiveEPCount field to 0, and not include the ActiveEPList field. If the NWKAddrOfInterest field does not match the network address of the remote device and it is the coordinator or a router, it SHALL determine whether the NWKAddrOfInterest field matches the network address of a device it holds in a discovery cache. If the NWKAddrOfInterest field does not match the network address of a device it holds in a discovery cache, it SHALL set the Status field to DEVICE_NOT_FOUND, set the ActiveEPCount

field to 0, and not include the ActiveEPList field. If the NWKAddrOfInterest matches the network address of a device held in a discovery cache on the remote device, it SHALL determine whether that device has any active endpoints. If the discovery information corresponding to the ActiveEP request has not yet been uploaded to the discovery cache, the remote device SHALL set the Status field to NO_DESCRIPTOR, set the ActiveEPCount field to 0 and not include the ActiveEPList field. If the cached device has no active endpoints, the remote device SHALL set the Status field to SUCCESS, set the ActiveEPCount field to 0, and not include the ActiveEPList field. If the cached device has active endpoints, the remote device SHALL set the Status field to SUCCESS, set the ActiveEPCount field to the number of active endpoints on that device, and include an ascending list of all the identifiers of the active endpoints on that device in the ActiveEPList field.

2.4.4.2.6.2 Effect on Receipt

On receipt of the Active_EP_rsp command, the recipient is either notified of the active endpoints of the remote device indicated in the original Active_EP_req command or notified of an error. If the Active_EP_rsp command is received with a Status of SUCCESS, the ActiveEPCount field indicates the number of entries in the ActiveEPList field. Otherwise, the Status field indicates the error and the ActiveEPList field SHALL NOT be included.

2.4.4.2.7 Match_Desc_rsp

The Match_Desc_rsp command (ClusterID=0x8006) SHALL be formatted as illustrated inFigure 2-62.

Octet: 1	2	1	Variable
Status	NWKAddrOfInterest	Match Length	Match List

Figure 2-62. Format of the Match_Desc_rsp Command Frame

Table 2-93 specifies the fields of the Match_Desc_rsp command frame.

Table 2-93. Fields of the Match_Desc_rsp Command Frame

Name	Type	Valid Range	Description
Status	Integer	SUCCESS, DEVICE_NOT_FOUND, INV_REQUESTTYPE, or NO_DESCRIPTOR	The status of the Match_Desc_req command.
NWKAddrOfInterest	Device Address	16-bit NWK address	NWK address for the request.
MatchLength	Integer	0x00-0xff	The count of endpoints on the Remote Device that match the request criteria.
MatchList			List of bytes each of which represents an 8-bit endpoint.

2.4.4.2.7.1 When Generated

The Match_Desc_rsp is generated by a remote device in response to a Match_Desc_req either broadcast or unicast to the remote device. This command SHALL be unicast to the originator of the Match_Desc_req command.

The following describes the procedure for processing the Match_Desc_req and generation of Match_Desc_rsp.

1. Set MatchLength to 0 and create an empty list MatchList.
2. If the receiving device is an End Device and the NWKAddrOfInterest within the Match_Desc_req message does not match the nwkNetworkAddress of the NIB and is not a broadcast address, the following SHALL be performed. Otherwise it shall proceed to step 3.
    a. If the NWK destination of the message is a broadcast address, no further processing SHALL be done.
    b. If the NWK destination is a unicast address, the following SHALL be performed.
        i. Set the Status value to INV_REQUESTTYPE.
        ii. Set the MatchLength to 0.
        iii. Construct a Match_Desc_rsp with only Status and MatchLength fields.
        iv. Send the message as a unicast to the source of the Match_Desc_req.
        v. No further processing SHALL be done.
3. If the NWKAddrOfInterest is equal to the nwkNetworkAddress of the NIB, or is a broadcast address, perform the following procedure. Otherwise proceed to step 4.
    a. Apply the match criteria in section 2.4.4.2.7.2 for all local Simple Descriptors.
    b. For each Simple Descriptor that matches with at least one cluster, add the endpoint once to MatchList and increment MatchLength.
4. If the NWKAddrOfInterest is not a broadcast address, the NWKAddressOfInterest is not equal to the nwkNetworkAddress of the local NIB, and the device is a coordinator or router, then the following SHALL be performed. Otherwise proceed to step 5.
    a. Examine each entry in the nwkNeighborTable and perform the following procedure.
        i. If the Network Address of the entry does not match the NWKAddrOfInterest or the Device Type is not equal to 0x02 (Zigbee End Device), do not process this entry. Continue to the next entry in the nwkNeighborTable.
        ii. For each endpoint that matches with at least once cluster, add that endpoint once to the MatchList and increment MatchLength.
        iii. Proceed to step 7.
    b. If the NWKAddrOfInterest does not match any entry in the nwkNeighborTable, perform the following:
        i. Set the Status to DEVICE_NOT_FOUND.
        ii. Construct a Match_Desc_rsp with Status and MatchLength fields only.
        iii. Unicast the message to the source of the Match_Desc_req.
        iv. No further processing SHALL be done.
5. If the MatchLength is 0 and the NWK destination of the Match_Desc_req was a broadcast address, no further processing SHALL be done. Otherwise proceed to step 6.
6. If the MatchLength is 0 and the NWKAddrOfInterest matched an entry in the nwkNeighborTable, the following SHALL be performed. Otherwise proceed to step 7.
    a. Set the Status to NO_DESCRIPTOR
    b. Construct a Match_Desc_rsp with Status and MatchLength only.
    c. Unicast the Match_Desc_rsp to the source of the Match_Desc_req.
    d. No further processing SHALL be done.
7. The following SHALL be performed. This is the case for both MatchLength > 0 and MatchLength == 0.
    a. Set the Status to SUCCESS.

b. Construct a Match_Desc_rsp with Status, NWKAddrOfInterest, MatchLength, and MatchList.

c. Unicast the response to the NWK source of the Match_Desc_req.

2.4.4.2.7.2 Simple Descriptor Matching Rules

These rules will examine a ProfileID, InputClusterList, OutputClusterList, and a SimpleDescriptor. The following SHALL be performed:

1. The device SHALL first check if the ProfileID field matches using the Profile ID of the SimpleDescriptor. If the profile identifiers do not match and the ProfileID is not 0xffff, the device SHALL note the match as unsuccessful and no further processing SHALL be done.
2. Examine the InputClusterList and compare each item to the Application Input Cluster List of the SimpleDescriptor.
   a. If a cluster ID matches exactly, then the device SHALL note the match as successful and perform no further matching. Processing is complete.
3. Examine the OutputClusterList and compare each item to the Application Output Cluster List of the SimpleDescriptor.
   a. If a cluster ID matches exactly, then the device SHALL note the match as successful and perform no further matching. Processing is complete.
4. The device SHALL note the match as unsuccessful. Processing is complete.

2.4.4.2.7.3 Effect on Receipt

On receipt of the Match_Desc_rsp command, the recipient is either notified of the results of its match criterion query indicated in the original Match_Desc_req command or notified of an error. If the Match_Desc_rsp command is received with a Status of SUCCESS, the MatchList field SHALL contain the list of endpoints containing simple descriptors that matched the criterion. Otherwise, the Status field indicates the error and the MatchList field SHALL NOT be included.

2.4.4.2.8 Complex_Desc_rsp – DEPRECATED

2.4.4.2.9 User_Desc_rsp – DEPRECATED

2.4.4.2.10 System_Server_Discovery_rsp

The System_Server_Discovery_rsp command (ClusterID=0x8015) SHALL be formatted as illustrated in Figure 2-63.

Octet: 1	2
Status	ServerMask

Figure 2-63. System_Server_Discovery_rsp Command Frame

Table 2-94 specifies the fields of the System_Server_Discovery_rsp command frame.

4477 **Table 2-94. Fields of the System_Server_Discovery_rsp Command Frame**

Name	Type	Valid Range	Description
Status	Integer	SUCCESS	The status of the System_Server_Discovery_rsp command.
ServerMask	Integer	Bitmap	See Table 2-34 for bit assignments.

4478 2.4.4.2.10.1 **When Generated**

4479 The System_Server_Discovery_rsp is generated from Remote Devices on receipt of a System_Server_
4480 Discovery_req primitive if the parameter matches the Server Mask field in its node descriptor. If there is no match,
4481 the System_Server_Discovery_req SHALL be ignored and no response given. Matching is performed by masking the
4482 ServerMask parameter of the System_Server_Discovery_req with the Server Mask field in the node descriptor. This
4483 command SHALL be unicast to the device which sent System_Server_Discovery_req with Acknowledge request set
4484 in TxOptions. The parameter ServerMask contains the bits in the parameter of the request which match the server
4485 mask in the node descriptor.

4486 2.4.4.2.10.2 **Effect on Receipt**

4487 The requesting device is notified that this device has some of the system server functionality that the requesting device
4488 is seeking.

4489 If the Network Manager bit was set in the System_Server_Discovery_rsp, then the Remote Device's NWK address
4490 SHALL be set into the *nwkManagerAddr* of the NIB.

4491 ### 2.4.4.2.11 User_Desc_conf – DEPRECATED

4492 ### 2.4.4.2.12 Discovery_Cache_rsp – DEPRECATED

4493 ### 2.4.4.2.13 Discovery_store_rsp – DEPRECATED

4494 ### 2.4.4.2.14 Node_Desc_store_rsp – DEPRECATED

4495 ### 2.4.4.2.15 Power_Desc_store_rsp – DEPRECATED

4496 ### 2.4.4.2.16 Active_EP_store_rsp– DEPRECATED

4497 ### 2.4.4.2.17 Simple_Desc_store_rsp – DEPRECATED

4498 ### 2.4.4.2.18 Find_node_cache_rsp – DEPRECATED

4499 ### 2.4.4.2.19 Extended_Simple_Desc_rsp – DEPRECATED

4500 ### 2.4.4.2.20 Extended_Active_EP_rsp – DEPRECATED

4501 ### 2.4.4.2.21 Remove_node_cache_rsp – DEPRECATED

4502 ### 2.4.4.2.22 Parent_annce_rsp

4503 The Parent_annce_rsp command (ClusterID = 0x801f) SHALL be formatted as illustrated in Figure 2-64, and is gen-
4504 erated in response to a Parent_annce.

4505

Octets: 1	1	Variable	…	Variable
Status	NumberOfChildren	ChildInfo[0]	…	ChildInfo[n]

Figure 2-64. Format of the Parent_annce_rsp Command Frame

Table 2-95 specifies the fields of the Parent_annce_rsp command frame.

Table 2-95. Fields of the Parent_annce_rsp Command Frame

Name	Type	Valid Range	Description
Status	Integer	SUCCESS or NOT_SUPPORTED	The status of the Parent_annce command.
NumberOfChildren	Integer	0 – 255	The number of ChildInfo structures contained in the message.
ChildInfo	ChildInfo	Variable	The child information. See Table 2-52.

Table 2-52 specifies the contents of the ChildInfo structure. This is the same format as the Parent_annce.

2.4.4.2.22.1 When Generated

Upon receipt of a Parent_annce message, a router SHALL construct but not yet send a Parent_annce_rsp message with the NumberOfChildren field set to 0. It SHALL then examine each Extended Address present in the Parent_annce message and search its Neighbor Table for an entry that matches. If a device is found and the Device Type is Zigbee end device (0x02), the router SHALL do the following.

1. If the Keepalive Received value is TRUE, it SHALL keep the parent/child relationship in the neighbor table unmodified. It SHALL then do the following:
    a. Append the ChildInfo structure to the Parent_annce_rsp.
    b. Increment NumberOfChildren by 1.
2. If the Keepalive Received value is FALSE, it SHALL remove the entry.

If the NumberOfChildren field value is 0, the local device SHALL discard the previously constructed Parent_Annce_rsp. No response message shall be sent.

If the NumberOfChildren field in the Parent_Annce_rsp is greater than 0, it SHALL unicast the message to the sender of the Parent_Annce message.

If the device has more ChildInfo entries than fit in a single message, it SHALL send additional messages. These messages do not have to be jittered or delayed since they are unicast to a single device. Each Parent_annce_rsp SHALL set the NumberOfChildren field to the number of entries contained within the message.

2.4.4.2.22.2 Effect on Receipt

On receipt of a Parent_annce_rsp, the device SHALL examine its Neighbor Table for each extended address in the ChildInfo entry and do the following.

i) If the entry matches and the Device Type is Zigbee End Device (0x02), it SHALL do the following:
    (1) Delete the entry from the Neighbor table.
ii) If the entry does not match, no more processing is performed on this ChildInfo entry.

There is no message generated in response to a Parent_annce_rsp.Bind, Unbind Bind Management Server Services.

Table 2-96 lists the commands supported by Device Profile: Bind and Unbind Server Services. Each of these primitives will be discussed in the following sections.

Table 2-96. Unbind and Bind Management Server Services Primitives

Bind and Unbind Server Service Commands	Cluster ID	Server Processing
End_Device_Bind_rsp	0x8020	Deprecated
Bind_rsp	0x8021	O
Unbind_rsp	0x8022	O
Bind_Register_rsp	0x8023	Deprecated
Replace_Device_rsp	0x8024	Deprecated
Store_Bkup_Bind_Entry_rsp	0x8025	Deprecated
Remove_Bkup_Bind_Entry_rsp	0x8026	Deprecated
Backup_Bind_Table_rsp	0x8027	Deprecated
Recover_Bind_Table_rsp	0x8028	Deprecated
Backup_Source_Bind_rsp	0x8029	Deprecated
Recover_Source_Bind_rsp	0x802a	Deprecated
Clear_All_Bindings_rsp	0x802b	O

2.4.4.2.23 End_Device_Bind_rsp – DEPRECATED

2.4.4.2.24 Bind_rsp

The Bind_rsp command (ClusterID=0x8021) SHALL be formatted as illustrated in Figure 2-65.

Octets: 1
Status

Figure 2-65. Format of the Bind_rsp Command Frame

Table 2-97 specifies the fields of the Bind_rsp command frame.

Table 2-97. Fields of the Bind_rsp Command Frame

Name	Type	Valid Range	Description
Status	Integer	SUCCESS, NOT_SUPPORTED, INVALID_EP, TABLE_FULL, or NOT_AUTHORIZED	The status of the Bind_req command.

2.4.4.2.24.1 When Generated

The Bind_rsp is generated in response to a Bind_req. If the Bind_req is processed and the Binding Table entry committed on the Remote Device, a Status of SUCCESS is returned. If the Remote Device is not a Primary binding table cache or the SrcAddress, a Status of NOT_SUPPORTED is returned. The endpoint of the Bind_req SHALL be checked to determine whether it is between the inclusive range of 0x01 to 0xFE, and if not a Bind_rsp SHALL be generated with a status of INVALID_EP.

2.4.4.2.24.2 Effect on Receipt

Upon receipt, error checking is performed on the request as described in the previous section. Assuming the Status is SUCCESS, the parameters from the Bind_req are entered into the Binding Table at the Remote Device via the APSME-BIND.request primitive.

2.4.4.2.25 Unbind_rsp

The Unbind_rsp command (ClusterID=0x8022) SHALL be formatted as illustrated in Figure 2-66.

Octets: 1
Status

Figure 2-66. Format of the Unbind_rsp Command Frame

Table 2-98 specifies the fields of the Unbind_rsp command frame.

Table 2-98. Fields of the Unbind_rsp Command Frame

Name	Type	Valid Range	Description
Status	Integer	SUCCESS, NOT_SUPPORTED, INVALID_EP, NO_ENTRY, or NOT_AUTHORIZED	The status of the Unbind_req command.

2.4.4.2.25.1 When Generated

The Unbind_rsp is generated in response to an Unbind_req. If the Unbind_req is processed and the corresponding Binding Table entry is removed from the Remote Device, a Status of SUCCESS is returned. If the Remote Device is not the Zigbee Coordinator or the SrcAddress, a Status of NOT_SUPPORTED is returned. The supplied endpoint SHALL be checked to determine whether it falls within the specified range. If it does not, a Status of INVALID_EP SHALL be returned. If the Remote Device is the Zigbee Coordinator or SrcAddress but does not have a Binding Table entry corresponding to the parameters received in the request, a Status of NO_ENTRY is returned.

2.4.4.2.25.2 Effect on Receipt

Upon receipt, error checking is performed on the response. If the status is SUCCESS, the device has successfully removed the binding entry for the parameters specified in the Unbind_req.

2.4.4.2.26 Bind_Register_rsp – DEPRECATED

2.4.4.2.27 Replace_Device_rsp – DEPRECATED

2.4.4.2.28 Store_Bkup_Bind_Entry_rsp – DEPRECATED

2.4.4.2.29 Remove_Bkup_Bind_Entry_rsp – DEPRECATED

2.4.4.2.30 Backup_Bind_Table_rsp – DEPRECATED

2.4.4.2.31 Recover_Bind_Table_rsp – DEPRECATED

2.4.4.2.32 Backup_Source_Bind_rsp – DEPRECATED

2.4.4.2.33 Recover_Source_Bind_rsp – DEPRECATED

2.4.4.2.34 Clear_All_Bindings_rsp

The Clear_All_Binding_rsp command (ClusterID=0x802b) SHALL be formatted as illustrated in Figure 2-67.

Octets: 1
Status

Figure 2-67. Format of the Clear_All_Bindings_rsp Command Frame

Table 2-99 specifies the fields of the Unbind_rsp command frame.

Table 2-99. Fields of the Clear_All_Bindings_rsp Command Frame

Name	Type	Valid Range	Description
Status	Integer	SUCCESS, NOT_SUPPORTED, NOT_AUTHORIZED, INV_REQUESTTYPE, or NO_MATCH.	The status of the ZDO Clear_All_Bindings_req.

2.4.4.2.34.1 When Generated

This command is generated in response to a ZDO Clear_All_Bindings_req.

2.4.4.2.34.2 Effect on Receipt

The receiver of this command learns the result of a previous ZDO Clear_All_Bindings_req.

2.4.4.3 Network Management Server Services

Table 2-100 lists the commands supported by Device Profile: Network Management Server Services. Each of these commands will be discussed in the following sections.

Table 2-100. Network Management Server Service Commands

Network Management Server Service Commands	Cluster ID	Server Processing
Mgmt_NWK_Disc_rsp	0x8030	Deprecated
Mgmt_Lqi_rsp	0x8031	M

Network Management Server Service Commands	Cluster ID	Server Processing
Mgmt_Rtg_rsp	0x8032	O
Mgmt_Bind_rsp	0x8033	O
Mgmt_Leave_rsp	0x8034	O
Mgmt_Direct_Join_rsp	0x8035	Deprecated
Mgmt_Permit_Joining_rsp	0x8036	M
Mgmt_Cache_rsp	0x8037	Deprecated
Mgmt_NWK_Update_notify	0x8038	O
Mgmt_NWK_Enhanced_Update_notify	0x8039	O
Mgmt_NWK_IEEE_Joining_List_rsp	0x803A	O
Mgmt_NWK_Unsolicited_Enhanced_Update_notify	0x803B	O
Mgmt_NWK_Beacon_Survey_rsp	0x803C	O

2.4.4.3.1 Mgmt_NWK_Disc_rsp – DEPRECATED COMMAND

2.4.4.3.2 Mgmt_Lqi_rsp

The Mgmt_Lqi_rsp command (ClusterID=0x8031) SHALL be formatted as illustrated in Figure 2-68.

Octets: 1	1	1	1	Variable
Status	NeighborTable Entries	Start Index	NeighborTable ListCount	NeighborTable List

Figure 2-68. Format of the Mgmt_Lqi_rsp Command Frame

Table 2-101 specifies the fields of the Mgmt_Lqi_rsp command frame.

Table 2-101. Fields of the Mgmt_Lqi_rsp Command Frame

Name	Type	Valid Range	Description
Status	Integer	NOT_SUPPORTED or any status code returned from the NLME-GET.confirm primitive.	The status of the Mgmt_Lqi_req command.

Name	Type	Valid Range	Description
NeighborTableEntries	Integer	0x00 – 0xff	Total number of Neighbor Table entries with unique addresses within the Remote Device.
StartIndex	Integer	0x00 – 0xff	Starting index within the Neighbor Table filtered on unique addresses to begin reporting for the NeighborTableList.
NeighborTableListCount	Integer	0x00 – 0x02	Number of Neighbor Table entries included within NeighborTableList.
NeighborTableList	List of Neighbor Descriptors	The list SHALL contain the number elements given by the NeighborTableListCount.	A list of descriptors, beginning with the StartIndex element and continuing for NeighborTableListCount, of the elements in the Remote Device's Neighbor Table including the device address and associated LQI (see Table 2-102 for details).

4597

4598 **Table 2-102. NeighborTableList Record Format**

Name	Size (Bits)	Valid Range	Description
Extended PAN Id	64	A 64-bit PAN identifier	The 64-bit extended PAN identifier of the neighboring device.
Extended address	64	An extended 64-bit, IEEE address	64-bit IEEE address that is unique to every device. If this value is unknown at the time of the request, this field SHALL be set to 0xffffffffffffffff.
Network address	16	Network address	The 16-bit network address of the neighboring device.
Device type	2	0x00 – 0x03	The type of the neighbor device: 0x00 = Zigbee coordinator 0x01 = Zigbee router 0x02 = Zigbee end device 0x03 = Unknown

Name	Size (Bits)	Valid Range	Description
RxOnWhenIdle	2	0x00 – 0x02	Indicates if neighbor's receiver is enabled during idle portions of the CAP: 0x00 = Receiver is off 0x01 = Receiver is on 0x02 = unknown
Affinity	3	0x00 – 0x03	The relationship between the neighbor and the current device: 0x00 = neighbor is the parent 0x01 = neighbor is a child 0x02 = neighbor is a sibling 0x03 = None of the above
Reserved	1		This reserved bit SHALL be set to 0.
Permit joining	2	0x00 - 0x02	An indication of whether the neighbor device is accepting join requests: 0x00 = neighbor is not accepting join requests 0x01 = neighbor is accepting join requests 0x02 = unknown
Reserved	6		Each of these reserved bits SHALL be set to 0.
Depth	8	0x00 – nwkcMaxDepth	The tree depth of the neighbor device. A value of 0x00 indicates that the device is the Zigbee coordinator for the network.
LQA	8	0x00 – 0xff	The estimated link quality for RF transmissions from this device. See section 3.6.3 for a discussion of how this is calculated.

2.4.4.3.2.1 When Generated

The Mgmt_Lqi_rsp is generated in response to an Mgmt_Lqi_req. If this management command is not supported, a status of NOT_SUPPORTED SHALL be returned and all parameter fields after the Status field SHALL be omitted. Otherwise, the Remote Device SHALL implement the following processing.

Upon receipt of and after support for the Mgmt_Lqi_req has been verified, the Remote Device SHALL perform an NLME-GET.request (for the *nwkNeighborTable* attribute) and process the resulting neighbor table (obtained via the NLME-GET.confirm primitive) to create the Mgmt_Lqi_rsp command. If *nwkNeighborTable* was successfully obtained but one or more of the fields required in the NeighborTableList record (see Table 2-102) are not supported (as they are optional), the Mgmt_Lqi_rsp SHALL return a status of NOT_SUPPORTED and all parameter fields after the Status field SHALL be omitted. Otherwise, the Mgmt_Lqi_rsp command SHALL contain the same status that was

contained in the NLME-GET.confirm primitive and if this was not SUCCESS, all parameter fields after the status field SHALL be omitted.

The Relationship field in the nwkNeighborTable entry maps to the Affinity field in the Mgmt_Lqi_rsp but with the following special processing. Routers SHALL report back the Relationship status in the Affinity field as follows. If the Relationship enumeration is 0x00 to 0x02, then the Affinity field SHALL be the same value. If the Relationship enumeration indicates 0x03 or greater, then the Affinity field SHALL be set to 0x03, None of the Above.

From the *nwkNeighborTable* attribute, the neighbor table SHALL be accessed, starting with the index specified by StartIndex, and SHALL be moved to the NeighborTableList field of the Mgmt_Lqi_rsp command. The entries reported from the neighbor table SHALL be those, starting with StartIndex and including whole NeighborTableList records (see Table 2-102) until the limit on MSDU size, i.e., *aMaxMACFrameSize* (see [B1]), is reached. Within the Mgmt_Lqi_rsp command, the NeighborTableEntries field SHALL represent the total number of Neighbor Table entries in the Remote Device. The parameter NeighborTableListCount SHALL be the number of entries reported in the NeighborTableList field of the Mgmt_Lqi_rsp command.

The extended address, device type, RxOnWhenIdle, and permit joining fields have “unknown” values which SHALL be returned where the values are not available.

2.4.4.3.2.2 Effect on Receipt

The local device is notified of the results of its attempt to obtain the neighbor table.

2.4.4.3.3 Mgmt_Rtg_rsp

The Mgmt_Rtg_rsp command (ClusterID=0x8032) SHALL be formatted as illustrated in Figure 2-69.

Octets: 1	**1**	**1**	**1**	**Variable**
Status	RoutingTable Entries	Start Index	RoutingTable ListCount	RoutingTable List

Figure 2-69. Format of the Mgmt_Rtg_rsp Command Frame

Table 2-103 specifies the fields of the Mgmt_Rtg_rsp command frame.

Table 2-103. Fields of the Mgmt_Rtg_rsp Command Frame

Name	**Type**	**Valid Range**	**Description**
Status	Integer	NOT_SUPPORTED or any status code returned from the NLME-GET.confirm primitive.	The status of the Mgmt_Rtg_req command.
RoutingTableEntries	Integer	0x00– 0xff	Total number of Routing Table entries within the Remote Device.
StartIndex	Integer	0x00– 0xff	Starting index within the Routing Table to begin reporting for the RoutingTable-List.
RoutingTableListCount	Integer	0x00– 0xff	Number of Routing Table entries included within RoutingTableList.

Name	Type	Valid Range	Description
RoutingTableList	List of Routing Descriptors	The list SHALL contain the number elements given by the Routing-TableListCount	A list of descriptors, beginning with the StartIndex element and continuing for RoutingTableListCount, of the elements in the Remote Device's Routing Table (see Table 2-104 for details).

Table 2-104. RoutingTableList Record Format

Name	Size (Bits)	Valid Range	Description
Destination address	16	The 16-bit network address of this route.	Destination address.
Status	3	The status of the route.	0x0=ACTIVE. 0x1=DISCOVERY_UNDERWAY. 0x2=DISCOVERY_FAILED. 0x3=INACTIVE. 0x4-0x7=Reserved.
Memory Constrained	1		A flag indicating whether the device is a memory constrained concentrator.
Many-to-one	1		A flag indicating that the destination is a concentrator that issued a many-to-one request.
Route record required	1		A flag indicating that a route record command frame SHOULD be sent to the destination prior to the next data packet.
Reserved	2		
Next-hop address	16	The 16-bit network address of the next hop on the way to the destination.	Next-hop address.

2.4.4.3.3.1 When Generated

The Mgmt_Rtg_rsp is generated in response to an Mgmt_Rtg_req. If this management command is not supported, a status of NOT_SUPPORTED SHALL be returned and all parameter fields after the Status field SHALL be omitted. Otherwise, the Remote Device SHALL implement the following processing.

Upon receipt of and after support for the Mgmt_Rtg_req has been verified, the Remote Device SHALL perform an NLME-GET.request (for the *nwkRouteTable* attribute) and process the resulting NLME-GET.confirm (containing the

nwkRouteTable attribute) to create the Mgmt_Rtg_rsp command. The Mgmt_Rtg_rsp command SHALL contain the same status that was contained in the NLME-GET.confirm primitive and if this was not SUCCESS, all parameter fields after the status field SHALL be omitted.

From the *nwkRouteTable* attribute, the routing table SHALL be accessed, starting with the index specified by StartIndex, and moved to the RoutingTableList field of the Mgmt_Rtg_rsp command. The entries reported from the routing table SHALL be those, starting with StartIndex and including whole RoutingTableList records (see Table 2-104) until MSDU size limit, that is, *aMaxMACFrameSize* (see [B1]), is reached. Within the Mgmt_Rtg_rsp command, the RoutingTableEntries field SHALL represent the total number of Routing Table entries in the Remote Device. The RoutingTableListCount field SHALL be the number of entries reported in the RoutingTableList field of the Mgmt_Rtg_req command.

2.4.4.3.3.2 Effect on Receipt

The local device is notified of the results of its attempt to obtain the routing table.

2.4.4.3.4 Mgmt_Bind_rsp

The Mgmt_Bind_rsp command (ClusterID=0x8033) SHALL be formatted as illustrated in Figure 2-70.

Octets: 1	**1**	**1**	**1**	**Variable**
Status	BindingTable Entries	Start Index	BindingTable ListCount	BindingTable List

Figure 2-70. Format of the Mgmt_Bind_rsp Command Frame

Table 2-105 specifies the fields of the Mgmt_Bind_rsp command frame.

Table 2-105. Fields of the Mgmt_Bind_rsp Command Frame

Name	**Type**	**Valid Range**	**Description**
Status	Integer	NOT_SUPPORTED or any status code returned from the APSME-GET.confirm primitive.	The status of the Mgmt_Bind_req command.
BindingTableEntries	Integer	0x00 – 0xff	Total number of Binding Table entries within the Remote Device.
StartIndex	Integer	0x00 – 0xff	Starting index within the Binding Table to begin reporting for the BindingTableList.
BindingTableListCount	Integer	0x00 – 0xff	Number of Binding Table entries included within BindingTableList.

Name	Type	Valid Range	Description
BindingTableList	List of Binding Descriptors	The list SHALL contain the number elements given by the BindingTableListCount.	A list of descriptors, beginning with the StartIndex element and continuing for BindingTableListCount, of the elements in the Remote Device's Binding Table (see Table 2-106 for details).

4656

4657 **Table 2-106. BindingTableList Record Format**

Name	Size (Bits)	Valid Range	Description
SrcAddr	64	A valid 64-bit IEEE address	The source IEEE address for the binding entry.
SrcEndpoint	8	0x01 – 0xfe	The source endpoint for the binding entry.
ClusterId	16	0x0000 – 0xffff	The identifier of the cluster on the source device that is bound to the destination device.
DstAddrMode	8	0x00 – 0xff	The addressing mode for the destination address. This field can take one of the non-reserved values from the following list: 0x00 = reserved 0x01 = 16-bit group address for DstAddr and DstEndpoint not present 0x02 = reserved 0x03 = 64-bit extended address for DstAddr and DstEndp present 0x04 – 0xff = reserved
DstAddr	16/64	As specified by the DstAddrMode field.	The destination address for the binding entry.
DstEndpoint	0/8	0x01 – 0xff	This field SHALL be present only if the DstAddrMode field has a value of 0x03 and, if present, SHALL be the destination endpoint for the binding entry.

4658 2.4.4.3.4.1 **When Generated**

4659 The Mgmt_Bind_rsp is generated in response to a Mgmt_Bind_req. If this management command is not supported, a
4660 status of NOT_SUPPORTED shall be returned and all parameter fields after the Status field shall be omitted. Other-
4661 wise, the Remote Device SHALL implement the following processing.

Upon receipt of and after support for the Mgmt_Bind_req has been verified, the Remote Device SHALL perform an APSME-GET.request (for the *apsBindingTable* attribute) and process the resulting APSME-GET.confirm (containing the *apsBindingTable* attribute) to create the Mgmt_Bind_rsp command. The Mgmt_Bind_rsp command SHALL contain the same status that was contained in the APSME-GET.confirm primitive and if this was not SUCCESS, all parameter fields after the status field SHALL be omitted. If the binding table is empty, the Mgmt_Bind_rsp SHALL return SUCCESS, set the fields BindingTable Entries = Start Index = BindingTable ListCount = 0x00 and not include the BindingTable List field.

From the *apsBindingTable* attribute, the binding table SHALL be accessed, starting with the index specified by StartIndex, and moved to the BindingTableList field of the Mgmt_Bind_rsp command. The entries reported from the binding table SHALL be those, starting with StartIndex and including whole BindingTableList records (see Table 2-106) until the MSDU size limit, that is,, *aMaxMACFrameSize* (see [B1]), is reached. Within the Mgmt_Bind_rsp command, the BindingTableEntries field SHALL represent the total number of Binding Table entries in the Remote Device. The BindingTableListCount field SHALL be the number of entries reported in the BindingTableList field of the Mgmt_Bind_req command.

2.4.4.3.4.2 Effect on Receipt

The local device is notified of the results of its attempt to obtain the binding table.

2.4.4.3.5 Mgmt_Leave_rsp

The Mgmt_Leave_rsp command (ClusterID=0x8034) SHALL be formatted as illustrated in Figure 2-71.

Octets: 1
Status

Figure 2-71. Format of the Mgmt_Leave_rsp Command Frame

Table 2-107 specifies the fields of the Mgmt_Leave_rsp command frame.

Table 2-107. Fields of the Mgmt_Leave_rsp Command Frame

Name	Type	Valid Range	Description
Status	Integer	NOT_SUPPORTED, NOT_AUTHORIZED or any status code returned from the NLME-LEAVE.confirm primitive.	The status of the Mgmt_Leave_req command.

2.4.4.3.5.1 When Generated

The Mgmt_Leave_rsp is generated in response to a Mgmt_Leave_req. Stacks certified prior to Revision 21 MAY or MAY NOT support this command. If this management command is not supported, a status of NOT_SUPPORTED SHALL be returned. All stacks certified to Revision 21 and later SHALL support this command.

2.4.4.3.5.2 Effect on Receipt

Upon receipt of the Mgmt_leave_rsp the device MAY parse the Status field to determine whether or not the remote device accepted the leave request.

2.4.4.3.6 Mgmt_Direct_Join_rsp – DEPRECATED

2.4.4.3.7 Mgmt_Permit_Joining_rsp

The Mgmt_Permit_Joining_rsp command (ClusterID=0x8036) SHALL be formatted as illustrated in Figure 2-72.

Octets: 1
Status

Figure 2-72. Format of the Mgmt_Permit_Joining_rsp Command Frame

Table 2-108 specifies the fields of the Mgmt_Permit_Joining_rsp command frame.

Table 2-108. Fields of the Mgmt_Permit_Joining_rsp Command Frame

Name	Type	Valid Range	Description
Status	Integer	SUCCESS, INV_REQUESTTYPE, NOT_AUTHORIZED, or any status code returned from the NLME-PERMIT-JOINING.confirm primitive.	The status of the Mgmt_Permit_Joining_rsp command.

2.4.4.3.7.1 When Generated

The Mgmt_Permit_Joining_rsp is generated in response to a unicast Mgmt_Permit_Joining_req. In the description which follows, note that no response SHALL be sent if the Mgmt_Permit_Joining_req was received as a broadcast to all routers. If this management command is not permitted by the requesting device, a status of INV_REQUESTTYPE SHALL be returned. Upon receipt and after support for Mgmt_Permit_Joining_req has been verified, the Remote Device SHALL execute the NLME-PERMIT-JOINING.request. The Mgmt_Permit-Joining_rsp SHALL contain the same status that was contained in the NLME-PERMIT-JOINING.confirm primitive.

2.4.4.3.7.2 Effect on Receipt

The status of the Mgmt_Permit_Joining_req command is notified to the requestor.

2.4.4.3.8 Mgmt_Cache_rsp – DEPRECATED

2.4.4.3.9 Mgmt_NWK_Update_notify

The Mgmt_NWK_Update_notify command (ClusterID=0x8038) SHALL be formatted as illustrated in Figure 2-73.

Octets: 1	4	2	2	1	Variable
Status	Scanned Channels	TotalTransmissions	TransmissionFailures	ScannedChannelsListCount	EnergyValues

Figure 2-73. Format of the Mgmt_NWK_Update_notify Command Frame

Table 2-109 specifies the fields of the Mgmt_NWK_Update_notify command frame.

4710 **Table 2-109. Fields of the Mgmt_NWK_Update_notify Command Frame**

Name	Type	Valid Range	Description
Status	Integer	SUCCESS, INV_REQUESTTYPE, NOT_SUPPORTED, or any status values returned from the MLME-SCAN.confirm primitive	The status of the Mgmt_NWK_Update_notify command.
ScannedChannels	Bitmap	0x00000000 – 0xffffffff.	The five most significant bits (b27,..., b31) represent the binary encoded Channel Page. The 27 least significant bits (b0, b1,... b26) indicate which channels were scanned (1 = scan, 0 = do not scan) for each of the 27 valid channels.
TotalTransmissions	Integer	0x0000 –0xffff	Count of the total transmissions reported by the device.
TransmissionFailures	Integer	x0000 –0xffff	Sum of the total transmission failures reported by the device.
ScannedChannelsListCount	Integer	0x00 – 0xff	The list SHALL contain the number of records contained in the EnergyValues parameter.
EnergyValues	Integer	List of ED values each of which can be in the range of 0x00 – 0xff.	The result of an energy measurement made on this channel in accordance with [B1].

4711 #### 2.4.4.3.9.1 When Generated

4712 The Mgmt_NWK_Update_notify is provided to enable Zigbee devices to report the condition on local channels to a
4713 network manager. The scanned channels list is the report of channels scanned and contains a count followed by a list
4714 of records, one for each channel scanned, each record including one byte of the energy level measured during the scan,
4715 or 0xff if there is too much interference on this channel.

4716 When sent in response to a Mgmt_NWK_Update_req command the status field SHALL represent the status of the
4717 request. This message SHALL NOT be sent unsolicited – use Mgmt_NWK_Unsolicited_Enhanced_Update_notify
4718 instead.

4719 #### 2.4.4.3.9.2 Effect on Receipt

4720 The local device is notified of the local channel conditions at the transmitting device, or of its attempt to update
4721 network configuration parameters.

4722 ### 2.4.4.3.10 Mgmt_NWK_Enhanced_Update_notify

4723 The Mgmt_NWK_Enhanced_Update_notify command (ClusterID=0x8039) SHALL be formatted as illustrated in
4724 Figure 2-74.

Octets: 1	4	2	2	1	Variable
Status	Scanned Channels	TotalTransmis-sions	Transmission-Failures	ScannedChan-nelsListCount	EnergyValues

Figure 2-74. Format of the Mgmt_NWK_Enhanced_Update_notify Command Frame

Table 2-110 specifies the fields of the Mgmt_NWK_Enhanced_Update_notify command frame.

Table 2-110. Fields of the Mgmt_NWK_Enhanced_Update_notify Command Frame

Name	Type	Valid Range	Description
Status	Integer	SUCCESS, INV_REQUESTTYPE, NOT_SUPPORTED, or any status values returned from the MLME-SCAN.confirm primitive.	The status of the Mgmt_NWK_Enhanced_Update_notify command.
ScannedChannels	Bitmap	0x00000000 – 0xffffffff.	The five most significant bits (b27,..., b31) represent the binary encoded Channel Page. The 27 least significant bits (b0, b1,... b26) indicate which channels were scanned (1 = scan, 0 = do not scan) for each of the 27 valid channels.
TotalTransmissions	Integer	0x0000 – 0xffff	Count of the total transmissions reported by the device.
TransmissionFailures	Integer	x0000 – 0xffff	Sum of the total transmission failures reported by the device.
ScannedChannelsList-Count	Integer	0x00 – 0xff	The list SHALL contain the number of records contained in the EnergyValues parameter.
EnergyValues	Integer	List of ED values each of which can be in the range of 0x00 – 0xff.	The result of an energy measurement made on this channel in accordance with [B1].

2.4.4.3.10.1 When Generated

The Mgmt_NWK_Enhanced_Update_notify is provided to enable Zigbee devices to report the condition on local channels to a network manager. The scanned channels list is the report of channels scanned and contains a count followed by a list of records, one for each channel scanned, each record including one byte of the energy level measured during the scan, or 0xff if there is too much interference on this channel.

When sent in response to a Mgmt_NWK_Enhanced_Update_req command the status field SHALL represent the status of the request. This message SHALL NOT be sent unsolicited – use Mgmt_NWK_Unsolicited_Enhanced_Update_notify instead.

2.4.4.3.10.2 Effect on Receipt

The local device is notified of the local channel conditions at the transmitting device, or of its attempt to update network configuration parameters.

2.4.4.3.11 Mgmt_NWK_IEEE_Joining_List_rsp

The Mgmt_NWK_IEEE_Joining_list_rsp command (Cluster ID=0x803A) SHALL be formatted as illustrated in Figure 2-75.

Octets: 1	0/1	0/1	0/1	0/1	0/1	0/Variable
Status	IeeeJoiningListUpdateID	JoiningPolicy	IeeeJoiningListTotal	StartIndex	IeeeJoiningCount	IeeeJoiningList

Figure 2-75. Format of the Mgmt_NWK_IEEE_Joining_List_rsp Command Frame

Table 2-111 specifies the fields of the Mgmt_NWK_IEEE_Joining_List_rsp command frame.

Table 2-111. Field Descriptions of the Mgmt_NWK_IEEE_Joining_List_rsp Command Frame

Name	Type	Valid Range	Description
Status	Integer	SUCCESS, INV_REQUESTTYPE, or NOT_SUPPORTED	The status of the Mgmt_NWK_IEEE_Joining_List_req command. If Status is not SUCCESS, no other fields are included.
IeeeJoiningListUpdateID	Integer	0x00 – 0xFF	The issue ID of the IeeeJoiningList. This field SHALL start at 0 and increment for each change to the IeeeJoiningList, or each change to the Joining Policywrapping to 0 after 0xFF.
JoiningPolicy	Enumeration	See Table 2-112.	This is an enumeration indicating one of the JoiningPolicy values allowed in Table 2-112.
IeeeJoiningListTotal	Integer	0x00 – 0xFF	The total number of IEEE Joining Addresses contained in the Mgmt_NWK_IEEE_Joining_List_rsp.
StartIndex	Integer	0x00 – 0xFF	The starting index in the mibIeeeJoiningList. This field SHALL be omitted if the IeeeJoiningListTotal is 0.
IeeeJoiningCount	Integer	x00 – 0xFF	The number of IEE joining messages contained in the ZDO message
IeeeJoiningList	List of IEEE values		A list of IEEE addresses from the mibIeeeJoiningList. This field SHALL be omitted if the IeeeJoiningListTotal is 0.

Table 2-112. ZDO JoiningPolicy Enumeration Values

Enumeration	Value	Description
ALL_JOIN	0x00	Any device is allowed to join.
IEEELIST_JOIN	0x01	Only devices on the mibJoiningIeeeList are allowed to join.
NO_JOIN	0x02	No device is allowed to join.

2.4.4.3.11.1 When Generated

The Mgmt_NWK_IEEE_Joining_List_rsp MAY either be generated in response to a Mgmt_NWK_IEEE_Joining_List_req or it MAY be sent as an unsolicited broadcast to inform the entire network of a change. For the details of when it is generated in response to a Mgmt_NWK_IEEE_Joining_List_req, see section 2.4.3.3.11.2.

2.4.4.3.11.2 Effect on Receipt

The device SHALL process the message as follows:

1) If the Status is not SUCCESS, the message SHALL be discarded and no further processing SHALL take place.
2) For each entry in the nwkMacInterfaceTable it SHALL do the following.
   a) Execute an MLME-SET.request of the *mibJoiningPolicy* to the value of the JoiningPolicy from the ZDO message.
   b) If the IeeeJoiningListTotal is 0 it SHALL do the following:
      i) The ZDO SHALL clear all entries from the *mibJoiningIeeeList*.
      ii) Go to step 2 and process the next entry in the nwkMacInterfaceTable.
   c) Execute an MLME-SET.request and set the values of the *mibJoiningIeeeList* at the index of StartIndex to the values of IeeeJoiningList from the ZDO message.

2.4.4.3.12 Mgmt_NWK_Unsolicited_Enhanced_Update_notify

The Mgmt_NWK_Unsolicited_Enhanced_Update_notify command (ClusterID=0x003b) SHALL be formatted as illustrated in Figure 2-76.

Octets: 1	4	2	2	2	1
Status	Channel in use	MACTxUcast Total	MACTxUcast Failures	MACTxUcast Retries	PeriodOfTimeForResults

Figure 2-76. Format of the Mgmt_NWK_Unsolicited_Update_notify Command Frame

Table 2-113 specifies the fields of the Mgmt_NWK_Unsolicited_Enhanced_Update_notify command frame.

Table 2-113 Fields of the Mgmt_NWK_Unsolicited_Enhanced_Update_notify Command

Name	Type	Valid Range	Description
Channel in use	Bitmap	0x00000000 – 0xffffffff	The five most significant bits (b27,..., b31) represent the binary encoded Channel Page. The 27 least significant bits (b0, b1,... b26) indicate which channels

Name	Type	Valid Range	Description
			is in use (1 = in use, 0 = not in use) for each of the 27 valid channels.
MACTxUcast Total	Integer	0x0000 –0xffff	Total number of Mac Tx Transactions to attempt to send a message (but not counting retries)
MACTxUcast Failures	Integer	x0000 – 0xffff	Total number of failed Tx Transactions. So if the Mac sent a single packet, it will be retried 4 times without ACK, that counts as 1 failure.
MACTxUcast Retries	Integer	x0000 – 0xffff	Total number of Mac Retries regardless of whether the transaction resulted in success or failure.
PeriodOfTimeForResults	Integer	0x00 – 0xff	Time period over which MACTxyyy results are measured (in minutes)

2.4.4.3.12.1 When Generated

The Mgmt_NWK_Unsolicited_Enhanced_Update_notify is provided to enable Zigbee devices to report the condition on local channels to a network manager. The scanned channel list is the report of channels scanned and it is followed by a list of records, one for each channel scanned, each record including one byte of the energy level measured during the scan, or 0xff if there is too much interference on this channel.

2.4.4.3.12.2 Effect on Receipt

The local device is notified of the local channel conditions at the transmitting device.

2.4.4.3.13 Mgmt_NWK_Beacon_Survey_rsp

The Mgmt_NWK_Beacon_Survey_rsp (ClusterID=0x803c) SHALL be formatted as illustrated in Figure 2-77.

Octets: 1	Varies
Status	TLVs

Figure 2-77. Format of the Mgmt_NWK_Beacon_Survey_rsp Command Frame

Table 2-114 specifies the fields of the Mgmt_NWK_Beacon_Survey_rsp command frame.

Table 2-114. Fields of the Mgmt_NWK_Beacon_Survey_rsp Command Frame

Name	Type	Valid Range	Description
Status	Integer	SUCCESS, INV_REQUESTTYPE, or NOT_SUPPORTED	The status of the Mgmt_NWK_Beacon_Survey_req command. If the status is not SUCCESS, then the other fields are not included.
TLVs	TLV	Varies	The Mgmt_NWK_Beacon_Survey_rsp SHALL include the following TLVs:

			• Beacon Survey Results TLV • Potential Parents TLV

4780 2.4.4.3.13.1 **Local TLVs**

4781 2.4.4.3.13.1.1 Beacon Survey Results TLV (ID 0x01)

4782 The Beacon Survey Results TLV (ID 0x01) is 4 bytes in length and contains information about the channels, scan
4783 configuration and counted beacons as illustrated in Figure 2-78.

Octets: 1	1	1	1
Total Beacons Received	On Network Beacon	Potential Parent Beacon	Other Network Beacons

4784 **Figure 2-78. Format of the Beacon Survey Results TLV**

4785 Table 2-115 specifies the fields of the Beacon Survey Results TLV.

4786 **Table 2-115. Fields of the Beacon Survey Results TLV**

Name	Type	Valid Range	Description
Total Beacons Received	Integer	0 – 255	The total number of IEEE Std 802.15.4 beacons received during the scan.
On-network Beacons	Integer	0 – 255	The total number of Zigbee Network beacons where the Extended PAN ID matches the local device's nwkExtendedPanId.
Potential Parent Beacons	Integer	0 – 255	The total number of Zigbee Network beacons where the Extended PAN ID matches and the Zigbee Beacon payload indicates End Device Capacity = TRUE.
Other Network Beacons	Integer	0 – 255	The total number of IEEE Std 802.15.4 beacons from other Zigbee networks or other IEEE Std 802.15.4 networks. Other Zigbee network beacons are defined as when the Extended PAN ID does not match the local Extended PAN ID.

4787 2.4.4.3.13.1.2 Potential Parents TLV(ID 0x02)

4788 The Potential Parents TLV(ID 0x02) is 4 to 19 bytes in length and indicates the number of available parents in radio
4789 range as illustrated in Figure 2-79. A maximum of 5 parents is supported for this TLV. The list of potential parents
4790 SHALL be ordered as described in section 3.6.1.5.2.

Octets: 2	1	1	0 / 2	0 / 1	Variable
Current Parent Short Address	LQA	Count	Potential Parent Short Address	LQA	Additional Potential Parent Short Address and LQA fields

4791 **Figure 2-79. Format of the Potential Parents TLV**

4792 Table 2-116 specifies the fields of the Potential Parents TLV.

4793 **Table 2-116. Fields of the Potential Parents TLV**

Name	Type	Valid Range	Description
Current Parent Short Address	Short Address	0x0000 – 0xFFFF	The short address that is the current parent for the device. For a router or coordinator this value SHALL be set to 0xFFFF.
LQA	Integer	0x00 – 0xFF	The value of the LQA of the current parent.
Count	Integer	0x00 – 0x05	This is the count of additional potential parent short addresses and their associated LQA. If there are no other potential parents this SHALL indicate 0. This value SHALL not be greater than 5.
Potential Parent Short Address	Short Address	0x0000 – 0xFFFF	The short address for a potential parent that the device can hear a beacon for.
LQA	Integer	0x00 – 0xFF	The LQA value of the associated potential parent.

4794 2.4.4.3.13.2 **When Generated**

4795 This is generated in response to the Mgmt_NWK_Beacon_Survey_req command.

4796 2.4.4.3.13.3 **Effect on Receipt**

4797 The application MAY use this to help manage the network.

4798 2.4.4.4 Security Server Services

4799 Table 2-117 lists the commands supported by the Device Profile related to Security Client services.

4800 **Table 2-117. Security Server Services**

Security Client Service	Cluster ID	Client Transmission	Server Processing
Security_Start_Key_Negotiation_rsp	0x8040	O	O
Security_Retrieve_Authentication_Token_rsp	0x8041	O	O
Security_Get_Authentication_Level_rsp	0x8042	M	M
Security_Set_Configuration_rsp	0x8043	M	M
Security_Get_Configuration_rsp	0x8044	M	M
Security_Start_Key_Update_rsp	0x8045	M	M
Security_Decommisioning_rsp	0x8046	M	M
Security_Challenge_rsp	0x8047	M	M

4801 2.4.4.4.1 Security_Start_Key_Negotiation_rsp

4802 The Security_Start_Key_Negotiation_rsp command (0x8040) shall be formatted as illustrated in Figure 2-80. This
4803 command SHALL NOT be APS encrypted.

When performing Key Negotiation with an unauthenticated neighbor that is not yet on the network, network layer encryption SHALL NOT be used on the message. If the message is being sent to unauthenticated device that is not on the network and is not a neighbor, it SHALL be relayed as described in section 4.6.3.7.7. Otherwise the message SHALL have network layer encryption.

Octets: 1	Variable
Status	TLVs

Figure 2-80. Format of the Security_Start_Key_Negotiation_rsp Command Frame

Table 2-118 specifies the fields of the Security_Start_Key_Negotation_rsp command frame.

Table 2-118. Fields of the Security_Start_Key_Negotation_rsp Command Frame

Name	Type	Valid Range	Description
Status	Integer	SUCCESS, INVALID_TLV, MISSING_TLV, TEMPORARY_FAILURE, NOT_AUTHORIZED	The result of the Security_Start_Key_Negotiation_req.
TLVs	TLV	Varies	The set of TLVs sent by the receiver of the Security_Start_Key_Negotiation_req.

2.4.4.4.1.1 Local TLVs

2.4.4.4.1.2 Curve25519 Public Point TLV (ID=0)

Figure 2-81 indicates the format of the Local TLV for Curve25519 Public Point TLV.

Octets: 8	32
Device EUI64	Public Point

Figure 2-81. Format of the Curve25519 Public Point TLV

Table 2-119 specifies the fields of the Curve25519 Public Point TLV

Table 2-119. Fields of the Curve25519 Public Point TLV

Field	Description
Device EUI64	This indicates the EUI64 of the device that generated the public point.
Public Point	The 32-byte Curve public point.

2.4.4.4.1.3 When Generated

The Security_Start_Key_Negotiation_rsp is generated after a device processes the Security_Start_Key_Negotiation_req and decides to reject the request, or after it has accepted the request and executed the corresponding cryptographic primitives. Typically, this is used to negotiate a Trust Center Link Key prior to becoming fully joined and authorized on a network, but it can be used after joining a network as well.

The security primitives for key negotiation are the APSME-KEY-NEGOTIATION primitives and are used by the stack to manage the process. See section 4.4.9 for more details. Their interaction with the over-the-air messages can be found in Figure 4-6.

When negotiating a Trust Center Link Key the device SHALL send at least the following TLV:

- Curve25519 Public Point TLV

2.4.4.4.1.4 Effect on Receipt

On receipt, the device SHALL do as follows:

1. If the Status is TEMPORARY_FAILURE, indicating that the current APSME-KEY-NEGOTIATION.request cannot be processed at the present time, the Stack SHOULD retry the operation by generating a new APSME-KEY-NEGOTIATION.request. The delay before initiating the retry SHALL be 5 seconds or greater.
2. If the Status is any other non-zero value then no further processing SHALL be done.
3. If more than one public point TLV is present then the message SHALL be dropped and no further processing SHALL be done.
4. If the Curve25519 Public Point TLV is not present, then the message SHALL be dropped and no more processing SHALL be done.
5. Generate an APSME-KEY-NEGOTIATE.confirm with the following parameters
   a. The PartnerLongAddress SHALL be set to the Device EUI64 within the Curve25519 Public Point TLV.
   b. The PublicPointData SHALL be set to the public point from the Curve25519 Public Point TLV.
   c. If the ZDO frame was contained within an APS Command Relay Message Upstream then it SHALL do the following:
      i. Set RelayCommand to TRUE.
      ii. Set RelayLongAddress to the address of the Device that sent the Network Data frame.

2.4.4.4.2 Security_Retrieve_Authentication_Token_rsp

The Security_Retrieve_Authentication_Token_rsp command SHALL be as illustrated in Figure 2-82.

Octets: 1	Variable
Status	TLVs

Figure 2-82. Format of the Security_Retrieve_Authentication_Token_rsp Command Frame

2.4.4.4.2.1 When Generated

This message is generated by the Trust Center as described in section 2.4.3.4.2.

2.4.4.4.2.2 Effect on Receipt

Upon receipt, the device SHALL do the following:

1. If the message was not APS encrypted by the Trust Center it SHALL be dropped and no further processing SHALL be done.
2. If the message was not sent by the Trust Center it SHALL be dropped and no further processing SHALL be done.
3. The device SHALL find the *apsDeviceKeyPairSet* entry associated with the Trust Center.
   a. If none is found, then the message SHALL be discarded and no further processing SHALL be done.
4. The device SHALL examine the PassphraseUpdateAllowed of the entry.
   a. If set to FALSE then the message SHALL be discarded and no further processing SHALL be done.
5. The device SHALL examine the TLVs and determine if there is a 128-bit Symmetric Passphrase Global TLV in the set.
   a. If none is present, then the message SHALL be discarded and no further processing SHALL be done.
6. The device SHALL copy the data of the 128-bit Symmetric Passphrase Global TLV to the value of the Passphrase value for the entry of the *apsDeviceKeyPairSet* AIB value.
7. The device SHALL set the PassphraseUpdateAllowed value of the entry to FALSE.

2.4.4.4.3 Security_Get_Authentication_Level_rsp

The Security_Get_Authentication_Level_rsp command (ClusterID= 0x8042) SHALL be formatted as illustrated in Figure 2-83.

Octets: 1	Variable
Status	TLVs

Figure 2-83. Format of the Security_Get_Authentication_Level_rsp Command Frame

Table 2-120 specifies the fields of the Security_Start_Key_Negotation_rsp command frame.

Table 2-120. Fields of the Security_Get_Authentication_Level_rsp Command Frame

Name	Type	Valid Range	Description
Status	Integer	SUCCESS, NOT_SUPPORTED, INV_REQUESTTYPE, MISSING_TLV, and NOT_AUTHORIZED	The status of the request to get the authentication level.
TLVs	TLVs	Varies	A list of one or more TLVs. The following TLVs have specified behavior in this Revision of the specification: • Device Authentication Level TLV Other TLVs may be included.

2.4.4.4.3.1 Local TLVs

2.4.4.4.3.2 Device Authentication Level TLV (ID=0)

The Device Authentication Level TLV is formatted as illustrated in Figure 2-84.

Octets: 8	1	1
IEEEAddrRemoteNode	InitialJoinMethod	AcitveLinkKeyType

Figure 2-84. Format of the Device Authentication TLV

Table 2-121 specifies the fields of the Device Authentication TLV.

Table 2-121. Fields of the Device Authentication TLV

Name	Type	Valid Range	Description
IEEEAddrRemoteNode	Device Address	An extended 64-bit, IEEE address	64-bit address for the node that is being inquired about.
InitialJoinMethod	Enumeration	0x00 – 0x03	This indicates the joining method that was used when the device joined the network. 0x00 = Anonymous

Name	Type	Valid Range	Description
			0x01 = Install Code Key 0x02 = Well-known Passphrase 0x03 = Install Code Passphrase
ActiveLinkKeyType	Enumeration	0x00 – 0x04	This indicates what Link Key update method was used to create the current active Link Key. 0x00 = Not Updated 0x01 = Key Request Method 0x02 = Unauthenticated Key Negotiation 0x03 = Authenticated Key Negotiation 0x04 = Application Defined Certificate Based Mutual Authentication

2.4.4.4.3.3 When Generated

The Security_Get_Authentication_Level_rsp is generated by a Remote Device in response to a Security_Get_Authentication_Level_req command inquiring as to the authentication level of the IEEEAddrOfInterest of an address held in the Key Pair Descriptor table. The destination addressing on this command SHALL be unicast. The command SHALL be APS encrypted.

2.4.4.4.3.4 Effect on Receipt

On receipt of the Security_Get_Authentication_Level_rsp command, the recipient is either notified of the status of its attempt to discover the current authentication level of an IEEE address or notified of an error. If the Security_Get_Authentication_Level_rsp command is received with a Status of SUCCESS, the remaining fields of the command contain the appropriate discovery information.

2.4.4.4.4 Security_Set_Configuration_rsp

The Security_Set_Configuration_rsp command (ClusterID=0x8043) SHALL be formatted as illustrated in Figure 2-85. The command contains a set of TLV Tag ID and TLV Processing Status pairs as defined by the TLV Status Count in Figure 2-86.

Octets: 1	Variable
Overall Status	TLVs

Figure 2-85. Security_Set_Configuration_rsp Command Frame

Table 2-122 specifies the fields of the Security_Set_Configuration_rsp command frame.

Table 2-122. Fields of the Security_Set_Configuration_rsp Command Frame

Name	Type	Valid Range	Description
Overall Status	Integer	SUCCESS, INV_REQUESTTYPE, or NOT_SUPPORTED	The overall status of a Security_Set_Configuration_req command.
TLVs	Variable	Varies	A set of one or more TLVs.

2.4.4.4.4.1 Local TLVs

2.4.4.4.4.1.1 Processing Status TLV (ID = 0)

The Processing Status TLV indicates the result of processing configuration changes from a set of TLVs sent in a previous message. The Processing Status TLV illustrated in Figure 2-86 will be 1 or more bytes in length and contain pairs of tag ID and processing status results, meaning it will always be an odd number in total length.

Octets: 1	0 or 1	0 or 1	0 or 1	0 or 1	…
TLV Status Count	Tag ID	Processing Status	Tag ID	Processing Status	…

Figure 2-86. Format of the Processing Status TLV

The Processing Status TLV contains a set of Tag ID and Processing Status results from a previous set of TLVs sent to the device to change its configuration. The TLV Status count will indicate the number of Tag ID and Processing Status pairs are present in the full TLV. The count may be zero, indicating that there were no known TLVs in the previous message that could be processed. When the TLV Status count is greater than 1, there SHALL be pairs of Tag ID and Processing Status values. For each pair, the tag ID will indicate a previously received TLV tag ID and the associated status of whether it is processed. The Processing Status value SHALL be one of the ZDP Enumerated Status values: SUCCESS, INV_REQUESTTYPE, or NOT_SUPPORTED.

2.4.4.4.4.2 When Generated

The Security_Set_Configuration_rsp is generated by a device in response to a Security_Set_Configuration_req. For each received TLV Tag ID in the Security_Set_Configuration_req there SHALL exist a TLV Tag ID and the corresponding TLV Processing Status of that TLV. If at least one TLV was successfully processed the Overall Status SHALL be SUCCESS.

2.4.4.4.4.3 Effect on Receipt

The device receiving this message can determine the results of a previous Security_Set_Configuration_req.

2.4.4.4.5 Security_Get_Configuration_rsp

The Security_Get_Configuration_rsp command (ClusterID = 0x08044) is generated by a device in response to a Security_Get_Configuration_req. For received Global TLV IDs in the prior request the device responds with its current state information as a list of TLVs contained in the Security_Get_Configuration_rsp. This command SHALL be APS encrypted. The format of the message is in Figure 2-87.

Octets: 1	Variable
Overall Status	TLVs

Figure 2-87. Security_Get_Configuration_rsp Command Frame

Table 2-123 specifies the fields of the Security_Get_Configuration_rsp command frame.

Table 2-123. Fields of the Security_Get_Configuration_rsp Command Frame

Name	Type	Valid Range	Description
Overall Status	Integer	SUCCESS, INV_REQUESTTYPE, or NOT_SUPPORTED	The overall status of a Security_Get_Configuration_req command.
TLVs	TLV	Variable	The value of the requested global TLV values.

2.4.4.4.5.1 Local TLVs

There are no Local TLVs defined for this message.

2.4.4.4.5.2 When Generated

This message is generated in response to the ZDO Security_Get_Configuration_req.

2.4.4.4.5.3 Effect on Receipt

The device can examine each received global TLV to learn the state of that TLV for the device sending the Security_Get_Configuration_rsp.

2.4.4.4.6 Security_Start_Key_Update_rsp

The Security_Start_Key_Update_rsp command (cluster ID = 0x8045) is formatted as illustrated in Figure 2-88. This command SHALL be APS encrypted.

Octets: 1
Status

Figure 2-88. Security_Start_Key_Update_rsp Command Frame

Table 2-124 specifies the fields of the Security_Start_Key_Update_rsp command frame.

Table 2-124. Fields of the Security_Start_Key_Update_rsp Command Frame

Name	Type	Valid Range	Description
Status	Integer	SUCCESS, INV_REQUESTTYPE, NOT_AUTHORIZED or NOT_SUPPORTED	The status of the request to Start the key update process.

2.4.4.4.6.1 When Generated

This is generated in response to a Security_Start_Key_Update_req.

2.4.4.4.6.2 Effect on Receipt

The Trust Center will learn the result of whether it's request ZDO Security_Start_Key_Update_req was successful.

2.4.4.4.7 Security_Decommission_rsp

The Security_Decomission_rsp is sent in response to a Security_Decomission_req to report the result of an attempt to decomission all data associated with a target EUI64. The command (cluster ID = 0x8046) is formatted as illustrated in Figure 2-89. This command SHALL be APS encrypted.

Octets: 1
Status

Figure 2-89. Security_Decommission_rsp Command Frame

Table 2-125 specifies the fields of the Security_Decommission_rsp command frame.

Table 2-125. Fields of the Security_Decommission_rsp Command Frame

Name	Type	Valid Range	Description
Status	Integer	SUCCESS, INV_REQUESTTYPE, NOT_AUTHORIZED or NOT_SUPPORTED	The status of the request to Start the key update process.

2.4.4.4.7.1 When Generated

This is generated in response to a Security_Decommission_req.

2.4.4.4.7.2 Effect on Receipt

The Trust Center will learn of the result of the decommissioning request of a third-party device by the sender of the Security_Decommission_rsp.

2.4.4.4.8 Security_Challenge_rsp

This command is used by a device to respond to a challenge and send its latest frame counter value to another device. The Security_Challenge_rsp (Cluster ID = 0x8047) is formatted as illustrated in Figure 2-90.

Octets: Varies
TLVs

Figure 2-90. Security_Challenge_rsp Command Frame

2.4.4.4.8.1 Locally Scoped TLVs

Table 2-126 defines the locally scoped TLVs for this message.

Table 2-126. Locally Scoped TLVs for Security_Challenge_rsp

Tag ID	Name
0x00	APS Frame Counter Response

2.4.4.4.8.2 APS Frame Counter Response TLV

Table 2-127 describes the format of the APS Frame Counter Response TLV.

Table 2-127. Format of the APS Frame Counter Response TLV

Octets: 8	8	4	4	8
Sender EUI64	Received Challenge Value	APS Frame Counter	Challenge Security Frame Counter	MIC

Table 2-128 describes the fields of the APS Frame Counter Response TLV.

Table 2-128 Fields of the APS Frame Counter Response TLV

Field	Description
Responder EUI64	The EUI64 of the device that is responding to the Security_Challenge_req with its own challenge.
Received Challenge Value	A randomly generated 64-bit value previously received in the APS Frame Counter Challenge TLV.
APS Frame Counter	The current outgoing APS security frame counter held by the Responder EUI64 device.
Challenge Security Frame Counter	The AES-CCM-128 outgoing frame counter used to generate the MIC over the octet sequence { tag || length || responder EUI-64 || received challenge value || APS frame counter } using the special nonce and AES-128 key for frame counter synchronization.
MIC	The AES-128-CCM 64-bit MIC (security level 2) on all previous fields of this TLV, excluding the challenge security frame counter, including Tag ID and length fields.

2.4.4.4.8.3 When Generated

This command is generated by a device responding to a ZDO Security_Challenge_req to inform the requester of the local device's current APS Frame counter.

This command SHALL NOT be APS encrypted.

2.4.4.4.8.4 Effect on Receipt

1. If the message was broadcast it SHALL be dropped and no further processing SHALL be done.
2. If the message did not include the APS Frame Counter Response TLV do the following.
    a. The message is dropped and no further processing SHALL be done.
3. If the Sender EUI64 does not match the *apsChallengeTargetEui64* then the message SHALL be dropped and no further processing SHALL be done.
4. If the apsChallengeValue of the AIB does not match the Challenge Value in the TLV, the message SHALL be dropped and no further processing SHALL be done.
5. Otherwise, follow the procedure in section .

2.4.5 ZDP Enumeration Description

This section explains the meaning of the enumerations used in the ZDP. Table 2-129 shows a description of the ZDP enumeration values.