# Z-Stack

# Monitor and Test API

Document Number: SWRA198

**Texas Instruments, Inc.**

San Diego, California USA

| Version | Description | Date |
|---------|-------------|------|
| 1.0 | Initial Release | 09/06/2008 |
| 1.1 | Update address type for AUTOPEND | 04/02/2009 |
| 1.2 | Add MT_AF command and MT_ZDO callback for source routing | 06/25/2009 |
| 1.3 | Add MT_AF commands to support inter-pan<br><br>Add MT_ZDO commands to support link key configuration<br><br>Add MT_ZDO commands to MSG callback register/remove/indicate | 01/17/2010 |
| 1.4 | Add MT_UTIL commands for link key establishment<br><br>Increase 'Length' field from uint8 to uint16 in:<br>MT_AF_DATA_REQUEST_EXT and MT_AF_INCOMING_MSG_EXT<br><br>Add MT_AF_DATA_STORE and MT_AF_DATA_RETRIEVE<br><br>Add MT_UTIL_DATA_REQ and UTIL_TEST_LOOPBACK<br><br>Move MT_MAC_SRC_MATCH commands to MT_UTIL | 07/08/2010 |
| 1.5 | Add MT_ZDO commands to support manual joining procedure:<br><br>  ▪ MT_ZDO_NWK_DISCOVERY_REQ<br><br>  ▪ MT_ZDO_JOIN_REQ<br><br>  ▪ MT_ZDO_NWK_DISCOVERY_CNF<br><br>  ▪ MT_ZDO_BEACON_NOTIFY_IND<br><br>  ▪ MT_JOIN_CNF<br><br>Add UTIL_APSME_LINK_KEY_NV_ID_GET command | 07/27/2010 |
| 1.6 | Add MT_SYS commands for handling system date and time:<br><br>  ▪ MT_SYS_GET_TIME<br><br>  ▪ MT_SYS_SET_TIME<br><br>Corrections to SYS_VERSION and ZDO_NODE_DESC_RSP commands<br><br>Add UTIL_APSME_REQUEST_KEY_CMD command.<br><br>Changed command name from ZDO_NETWORK_DISCOVERY_REQ to ZDO_NWK_DISCOVERY_REQ<br><br>Fixed Cmd0 for ZDO_NWK_DISCOVERY_REQ and ZDO_JOIN_REQ.<br><br>Add AF_APSF_CONFIG_SET, SYS_OSAL_NV_ITEM_INIT, SYS_OSAL_NV_ITEM_LENGTH, SYS_OSAL_NV_ITEM_DELETE commands. | 07/16/2011 |
| 1.7 | Add ZDO_LEAVE_IND callback | 03/10/2012 |

| | | |
|---|---|---|
| | Add SYS_SET_TX_POWER command<br><br>Fixed parameter info for UTIL_ADDRMGR_EXT_ADDR_LOOKUP | |
| 1.8 | Add Wildcard Profile ID bit definition in AF_DATA_REQUEST and AF_DATA_REQUEST_EXT<br><br>Add AF_REFLECT_ERROR message<br><br>Clarify RemoveChildren_Rejoin parameter for ZDO_MGMT_LEAVE_REQ command | 11/14/2012 |
| 1.9 | Add UTIL_SRNG_GEN and UTIL_BIND_ADD_ENTRY commands<br>Modify ZDO_MGMT_PERMIT_JOIN_REQ command, added AddrMode field. | 11/23/2013 |
| 1.10 | Add MT_SYS commands for statistics:<br>  ▪ SYS_ZDIAGS_INIT_STATS<br><br>  ▪ SYS_ZDIAGS_CLEAR_STATS<br><br>  ▪ SYS_ZDIAGS_GET_STATS<br><br>  ▪ SYS_ZDIAGS_RESTORE_STATS_NV<br><br>  ▪ SYS_ ZDIAGS_SAVE_STATS_TO_NV | 03/13/2014 |
| 1.11 | Correction to UTIL_KEY_EVENT command | 12/09/2014 |
| 1.12 | Added NV MT commands<br>  • SYS_NV_CREATE<br>  • SYS_NV_DELETE<br>  • SYS_NV_READ<br>  • SYS_NV_WRITE<br>  • SYS_NV_UPDATE<br>  • SYS_NV_COMPACT<br>Added ZDO command for new rejoin parameter ZDO_SET_REJOIN_PARAMETERS_REQ<br>Added MT_ZDO extension commands<br>  • MT_ZDO_SEC_ADD_LINK_KEY<br>  • MT_ZDO_SEC_ENTRY_LOOKUP_EXT<br>  • MT_ZDO_SEC_DEVICE_REMOVE<br>  • MT_ZDO_EXT_ROUTE_DISC<br>  • MT_ZDO_EXT_ROUTE_CHECK<br>  • MT_ZDO_EXT_REMOVE_GROUP<br>  • MT_ZDO_EXT_REMOVE_ALL_GROUP<br>  • MT_ZDO_EXT_FIND_ALL_GROUPS_ENDPOINT<br>  • MT_ZDO_EXT_FIND_GROUP<br>  • MT_ZDO_EXT_ADD_GROUP | 02/19/2015 |

| | | |
|---|---|---|
| | • MT_ZDO_EXT_COUNT_ALL_GROUPS <br> • MT_ZDO_EXT_RX_IDLE <br> • MT_ZDO_EXT_UPDATE_NWK_KEY <br> • MT_ZDO_EXT_SWITCH_NWK_KEY <br> • MT_ZDO_EXT_NWK_INFO <br> • MT_ZDO_EXT_SEC_APS_REMOVE_REQ <br> • MT_ZDO_FORCE_CONCENTRATOR_CHANGE <br> • MT_ZDO_EXT_SET_PARAMS | |
| 1.13 | Added MT_ZDO_TC_DEVICE_IND, MT_ZDO_PERMIT_JOIN_IND , MT_ZDO_NWK_ADDR_OF_INTEREST_REQ commands | 03/09/2015 |
| 1.14 | Addition of new MT commands for the Z3 stack update <br> • New ZDO command ZDO_STARTUP_FROM_APP_EX <br> Added MT_APP_CNF commands <br> • APP_CNF_SET_NWK_FRAME_COUNTER <br> • APP_CNF_SET_DEFAULT_REMOTE_ENDDEVICE _TIMEOUT <br> • APP_CNF_SET_ENDDEVICETIMEOUT <br> • APP_CNF_SET_ALLOWREJOIN_TC_POLICY <br> • APP_CNF_BDB_START_COMMISSIONING <br> • APP_CNF_BDB_SET_CHANNEL <br> • APP_CNF_BDB_ADD_INSTALLCODE <br> • APP_CNF_BDB_SET_TC_REQUIRE_KEY_EXCHANGE <br> • APP_CNF_BDB_SET_JOINUSESINSTALLCODEKEY <br> • APP_CNF_BDB_SET_ACTIVE_DEFAULT_CENTRALIZED <br> •  APP_CNF_BDB_COMMISSIONING_NOTIFICATION <br> • APP_CNF_BDB_ZED_ATTEMPT_RECOVER_NWK <br><br> Added MT_GREENPOWER commands <br> • GP_DATA_REQ <br> • GP_SEC_RSP <br> • GP_DATA_CNF <br> • GP_SEC_REQ <br> • GP_DATA_IND | 10/25/2016 |
| 1.15 | MT_APP_CNF <br> • ZDO_STARTUP_FROM_APP_EX deprecated. ZDO_STARTUP_FROM_APP fixed to be backward compatible. <br> • BDB_COMMISSIONING_NOTIFICATION remaining methods parameter now returns the right value. <br> MT_UTIL <br> • UTIL_GET_DEVICE_INFO now returns Reduce Function Devices associated to the local device instead of | 5/10/2017 |

| | all devices in the association table. | |
|---|---|---|
| 1.16 | MT_APP_CNF<br><br>• APP_CNF_BDB_SET_ACTIVE_DEFAULT_CENTRALIZED_KEY updated to support fallback to default global centralized key and text plain key modes. | 6/21/2018 |
| 1.17 | Update to GP_DATA_IND command<br>Update SYS_SET_TX_POWER SRSP attribute | 5/23/2019 |

# Table of Contents

# 1.    Introduction

## 1.1    Scope

This document describes the Monitor and Test (MT) interface that is used for communication between the host tester and a ZigBee device through RS-232 serial port. Tester can issue MT commands to the ZigBee target through a PC application called Z-Tool. The target must be programmed with the latest Texas Instruments Z-Stack™.

## 1.2    Overview

MT interfaces are divided into categories, shown in the table below. Most interfaces can be disabled or enabled by a compile flag. Depending on the desired interfaces, certain flags need to be enabled during compilation. For a list of supported compile flags, check the "**Z-Stack Compile Option**" document.

| Interface | Description | Compile flags |
|---|---|---|
| **MT_AF** | This interface allows tester to interact with AF layer of the target. | **MT_AF_FUNC** <br> **MT_AF_CB_FUNC** |
| **MT_APP** | This interface allows tester to interact with APP layer of the target to control custom tests such as test profile or user-defined test. | **MT_APP_FUNC** |
| **MT_DEBUG** | This interface allows tester to control the debug-messaging mechanism such as debug threshold, debug messages, etc. | **MT_DEBUG_FUNC** |
| **MT_NWK** | This interface allows tester to interact with NWK layer of the target. | **MT_NWK_FUNC** <br> **MT_NWK_CB_FUNC** |
| **MT_SAPI** | This interface allows tester to interact with simple API interface. | **MT_SAPI_FUNC** <br> **MT_SAPI_CB_FUNC** |
| **MT_SYS** | This interface allows the tester to interact with the target at system level such as reset, read/write memory, read/write extended address, etc. | **MT_SYS_FUNC** |
| **MT_TASK** | This interface handles communication between the Monitor Test interface and Z-Stack. Tester has no control direct over this interface. | **MT_TASK** |
| **MT_UART** | This interface handles communication between the target and Z-Tool. Tester has no direct control over this interface. | **N/A** |
| **MT_UTIL** | This interface provides tester supporting functionalities such as setting PanID, getting device info, getting NV info, subscribing callbacks, etc. | **MT_UTIL_FUNC** |
| **MT_VERSION** | This interface contains information about the release version of the software. | **N/A** |

| | | |
|---|---|---|
| **MT_ZDO** | This interface allows tester to interact with the ZDO layer of the target. | **MT_ZDO_FUNC**  **MT_ZDO_CB_FUNC** |
| **MT_APP_CNF** | This interface allows to interact with the commands introduced by Z3.0 Base Device Behavior. | **MT_APP_CNF_FUNC** |
| **MT_GP** | This interface allows to interact with GP stubs. | **MT_GP_CB_FUNC** |

## 1.3    Requirements

There are several requirements for a tester to interact with the ZigBee target through the MT interface:

- ZigBee target is programmed with Texas Instruments Z-Stack™ (ZStack-2.1.0 or newer).

- Z-Tool 2.0 or newer installed on the tester PC.

- PC is connected to ZigBee target though RS-232 serial port.

## 1.4    Acronyms

**Table 1**

| | |
|---|---|
| **ADC** | Analog to Digital Conversion |
| **AF** | Application Framework |
| **AREQ** | Asynchronous Request |
| **FCS** | Frame Check Sequence |
| **MT** | Monitor and Test |
| **RPC** | Remote Procedure Call |
| **SAPI** | Simple API |
| **SOF** | Start of Frame |
| **SREQ** | Synchronous Request |
| **SRSP** | Synchronous Response |
| **Z-Stack** | Texas Instruments ZigBee protocol stack |
| **Z-Tool** | Texas Instruments ZigBee PC-based test tool |

## 1.5    Reference Documents

[1] Z-Stack Compile Options (SWRA188).

[2] Z-Stack User's Guides (SWRA161, SWRA162, SWRA163, SWRA164, SWRA165)

[3] Z-Stack Developer's Guide (SWRA176)

[4] Z-Stack Application Programming Interface (SWRA195)

## 2.      Monitor and Test Transport Protocol

- A transport protocol is necessary so that messages can be exchanged between the tester and target over an RS-232 serial link. The purpose of the transport protocol is to frame the messages in packets for proper transmission and reception and to ensure message integrity.

- The physical transmission uses:  no Parity; 8 data bits and 1 stop bits for each byte.

- The transmission rate will be 38.4 kbps, 57.6kbps and 115.2kbps

- The Z-Tool program must send one message at a time and wait for either the expected response message to a timeout before sending the next message or resending the current message.

- Fields that are multi-byte fields are transmitted Least Significant byte first (LSB). There is no provision for retransmission of lost packets

### 2.1      Format

### 2.1.1   General Serial Packet

- Serial packets are sent between the Z-Tool PC application and the target ZigBee device. They contain an SOF (Start of Frame), followed by a variable-length MT packet, and terminated by an FCS (Frame Check Sequence).

- Building of the serial packets is handled by MT_TransportSend() where the SOF is inserted at the beginning of the packet and FCS is computed and appended to the end of the packet.

| SOF | MT CMD | FCS |
|---|---|---|
| **Byte: 1** | **3-256** | **1** |

**SOF** (Start of Frame): This is a one byte field with value equal to **0xFE** that defines the start of each general serial packet.

**MT CMD** (Monitor Test Command): This contains 1 byte for the length of the actual data, 2 bytes for the MT command Id, and the data ranging from 0-250 bytes. Check 2.1.2 for more details.

**FCS** (Frame Check Sequence): This is a one byte field that is used to ensure packet integrity. This field is computed as an XOR of all the bytes in the message starting with LEN field and through the last byte of data. The receiver XORs all the received data bytes as indicated above and then XORs the received FCS field. If the sum is not equal to zero, the received packet is in error.

## 2.1.2   MT CMD

- MT CMD is the actually Monitor and Test command. It contains information that Z-Tool and Z-Stack need to control the target.

- It contains 1 byte for the length of the actual data, 2 bytes for the command, and data ranging from 0-250 bytes.

| LEN | CMD | DATA |
|---|---|---|
| Byte: 1 | 2 | 0-250 |

**LEN** (Length):  This one byte field is the number of bytes in the **DATA** field.  If the **DATA** field contains no information this LEN field has a value of 0 and the total length of the **MT CMD** is 3 bytes (0 data message).

**CMD** (Command Id): This is a two byte field with a value denoting the Command Identification (Id) for this message.  This field is described in detail below.

| CMD0 | | CMD1 |
|---|---|---|
| Bit: 7-5 | 4-0 | 7-0 |
| Type | Subsystem | Id |

**Type**:  Type for the command is described by bit 5, 6, 7 of CMD0 byte. The command type has one of the following values:

| Type | CMD0Value |
|---|---|
| POLL | 0x00 |
| SREQ | 0x20 |
| AREQ | 0x40 |
| SRSP | 0x60 |

- 0:  POLL.  A POLL command is used to retrieve queued data.  This command is only applicable to SPI transport.  For a POLL command the subsystem and Id are set to zero and data length is zero.

- 1: SREQ: A synchronous request that requires an immediate response. For example, a function call with a return value would use an SREQ command.

- 2: AREQ: An asynchronous request. For example, a callback event or a function call with no return value would use an AREQ command.

- 3: SRSP: A synchronous response. This type of command is only sent in response to a SREQ command. For an SRSP command the subsystem and Id are set to the same values as the corresponding SREQ. The length of an SRSP is generally nonzero, so an SRSP with length=0 can be used to indicate an error.

- 4-7: Reserved.

**Subsystem**: The subsystem of the command is described by bit 0, 1, 2, 3, 4 of CMD0. The command subsystem has one of the following values:

| Subsystem | Subsystem Value |
|---|---|
| Reserved | 0x00 |
| SYS interface | 0x01 |
| MAC interface | 0x02 |
| NWK interface | 0x03 |
| AF interface | 0x04 |
| ZDO interface | 0x05 |
| SAPI interface | 0x06 |
| UTIL interface | 0x07 |
| DEBUG interface | 0x08 |
| APP interface | 0x09 |
| APP config | 0x0F |
| GreenPower | 0x15 |

**Id**: The command Id. The Id maps to a particular interface message. Range: 0-250.

**DATA**: This field contains the actual data to be transmitted. This is a field which varies in size according to the command. It can be 0 to 250.

## 2.2    Example

**SYS_PING** command will look like **0xFE 0x00 0x21 0x01 0x20**

| SOF | LEN | CMD0 | CMD1 | DATA | FCS |
|------|------|------|------|------|------|
| Byte: 1 | 1 | 1 | 1 | 0 | 1 |
| 0xFE | 0x00 | 0x21 | 0x01 | N/A | 0x20 |

**SYS_PING** response will look like **0xFE 0x02 0x61 0x01 0x11 0x00 0x73**

| SOF | LEN | CMD0 | CMD1 | DATA0 | DATA1 | FCS |
|------|------|------|------|------|------|------|
| Byte: 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0xFE | 0x02 | 0x61 | 0x01 | 0x11 | 0x00 | 0x73 |

## 3.    Monitor and Test Commands

### 3.1    Introduction

Monitor and Test commands (MT commands) exchanged between the target and the tester via a supported H/W medium (i.e.RS-232 or USB.) The tester controls the target using Z-Tool 2.0. In order for the target to communicate with Z-Tool 2.0, Z-Stack must be compiled with MT_SYS_FUNC. This enables the MT_SYS interface so Z-Tool 2.0 can communicate to establish the connection. Some MT interfaces support callbacks. This requires MT_UTIL_FUNC to be compiled with Z-Stack in order for the tester to subscribe callback. The corresponding MT interface must also be complied with the correct flag in order for the callbacks to be received and processed correctly by Z-Stack and Z-Tool 2.0. For the complete details on MT flags, check section 1.2 or "Z-Stack Compile Option" document (SWRA188).

Summary:

- Z-Tool 2.0 installed and connected to target using the supported H/W interface.

- Z-Stack must be compiled with MT_SYS_FUNC and MT_UTIL_FUNC.

- Z-Stack must be compiled with MT interface what tester will use.

- Z-Stack and Z-Tool must be set at the same baud rate, no Parity, 8 data-bits and 1 stop-bit for each byte.

- If  the target supports flow control, this must be set correctly as well in Z-Tool 2.0

### 3.2    MT_AF

This interface allows the tester to interact with the Application Framework layer (AF).

### 3.2.1   MT_AF Commands

#### 3.2.1.1     AF_REGISTER

**Description**:

This command enables the tester to register an application's endpoint description.

**Usage**:

**SREQ:**

| 1 | 1 | 1 | 1 | 2 | 2 |
|---|---|---|---|---|---|
| Length = 0x09-0x49 | Cmd0 = 0x24 | Cmd1 = 0x00 | EndPoint | AppProfId | AppDeviceId |

| 1 | 1 | 1 | 0-32 | 1 | 0-32 |
|---|---|---|---|---|---|
| AppDevVer | LatencyReq | AppNumInClusters | AppInClusterList | AppNumOutClusters | AppOutClusterList |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| EndPoint | 1 | Specifies the endpoint of the device |
| AppProfId | 2 | Specifies the profile Id of the application |
| AppDeviceId | 2 | Specifies the device description Id for this endpoint |
| AddDevVer | 1 | Specifies the device version number |
| LatencyReq | 1 | Specifies latency. 0x00-No latency 0x01-fast beacons 0x02-slow beacons |
| AppNumInClusters | 1 | the number of Input cluster Id's following in the AppInClusterList |
| AppInClusterList | 32 | Specifies the list of Input Cluster Id's |
| AppNumOutClusters | 1 | Specifies the number of Output cluster Id's following in the AppOutClusterList |
| AppOutClusterList | 32 | Specifies the list of Output Cluster Id's |

**SRSP:**

| Byte:1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x64 | Cmd1 = 0x00 | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.2.1.2    AF_DATA_REQUEST
**Description**:

This command is used by the tester to build and send a message through AF layer.

**Usage**:

**SREQ:**

| Byte: 1 | 1 | 1 | 2 | 1 |
|---|---|---|---|---|
| Length = 0x0A-0x8A | Cmd0 = 0x24 | Cmd1 = 0x01 | DstAddr | DstEndpoint |

| Byte: 1 | 2 | 1 | 1 | 1 | 1 | 0-128 |
|---|---|---|---|---|---|---|
| SrcEndpoint | ClusterId | TransId | Options | Radius | Len | Data |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| DstAddr | 2 | Short address of the destination device |
| DstEndpoint | 1 | Endpoint of the destination device |
| SrcEndpoint | 1 | Endpoint of the source device |
| ClusterId | 2 | Specifies the cluster ID |
| TransId | 1 | Specifies the transaction sequence number of the message. |
| Options | 1 | Transmit options bit mask according to the following defines from AF.h: bit 1: sets 'Wildcard Profile ID'; bit 4: turns on/off 'APS ACK'; bit 5 sets 'discover route'; bit 6 sets 'APS security'; bit 7 sets 'skip routing'. |
| Radius | 1 | Specifies the number of hops allowed delivering the message (see AF_DEFAULT_RADIUS.) |
| Len | 1 | Length of the data. |
| Data | 0-128 | 0-128 bytes data |

**SRSP:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x64 | Cmd1 = 0x01 | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.2.1.3     AF_DATA_REQUEST_EXT

**Description**:

This extended form of the AF_DATA_REQUEST must be used to send an inter-pan message (note that the target code must be compiled with the INTER_PAN flag defined.) This extended data request must also be used when making a request with a huge data byte count which is defined to be a size that would cause the RPC request to exceed the maximum allowed size:

```
               MT_RPC_DATA_MAX – sizeof(AF_DATA_REQUEST_EXT)
```

Where `sizeof(AF_DATA_REQUEST_EXT)` counts everything but the data bytes and now stands at 20. When making an AF_DATA_REQUEST_EXT with a huge data byte count, the request shall not contain any data bytes. The huge data buffer is sent over separately as a sequence of one or more AF_DATA_STORE requests. Note that the outgoing huge message is timed-out in 15 seconds; thus all AF_DATA_STORE requests must be completed within 15 seconds of an AF_DATA_REQUEST_EXT with a huge data byte count. And any AF_DATA_REQUEST_EXT with a huge data byte count must be completed (or timed-out) before another will be started. The default timeout can be changed by defining the following to other values:

```
#if !defined MT_AF_EXEC_CNT

#define MT_AF_EXEC_CNT  15

#endif

#if !defined MT_AF_EXEC_DLY

#define MT_AF_EXEC_DLY  1000

#endif
```

**Usage**:

**SREQ:**

| 1 | 1 | 1 | 1 | 8 | 1 |
|---|---|---|---|---|---|
| Length = 0x14-0x93 | Cmd0 = 0x24 | Cmd1 = 0x02 | DstAddrMode | DstAddr | DstEndpoint |

| 2 | 1 | 2 | 1 | 1 | 1 | 2 | 0-128 |
|---|---|---|---|---|---|---|---|
| DstPanId | SrcEndpoint | ClusterId | TransId | Options | Radius | Len | Data |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| DstAddrMode | 1 | A value of 3 (the enumeration value for 'afAddr64Bit') indicates 8-byte (64-bit) address mode; otherwise a value of 2 indicates 2-byte (16-bit) address mode, using only the 2 LSB's of the DstAddr field to form a 2-byte short address. |
| DstAddr | 8 | LSB to MSB for the long or short address of the destination device (upper 6 bytes are don't care when short address.) |
| DstEndpoint | 1 | Endpoint of the destination device (but don't care if the DstPanId is non-zero, which indicates an inter-pan message.) |
| DstPanId | 2 | PanId of the destination device: 0x0000=Intra-Pan; otherwise, Inter-Pan. |
| SrcEndpoint | 1 | Endpoint of the source device. |
| ClusterId | 2 | Specifies the cluster ID |
| TransId | 1 | Specifies the transaction sequence number of the message. |
| Options | 1 | Transmit options bit mask according to the following defines from AF.h: bit 1: sets 'Wildcard Profile ID'; bit 4: turns on/off 'APS ACK'; bit 5 sets 'discover route'; bit 6 sets 'APS security'; bit 7 sets 'skip routing'. (This doesn't apply for |

|  |  |  |
|---|---|---|
|  |  | an inter-pan message.) |
| Radius | 1 | Specifies the number of hops allowed delivering the message (reference DEF_NWK_RADIUS.) |
| Len | 2 | Length of the data. If a large data length causes the MT command to exceed MT_RPC_DATA_MAX, then zero bytes of the data shall be sent with this request and the data shall be transferred in as many MT_AF_DATA_STORE requests as necessary. |
| Data | 0-128 | 0-128 bytes data |

**SRSP:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x64 | Cmd1 = 0x02 | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.2.1.4    AF_DATA_REQUEST_SRC_RTG

**Description**:

This command is used by the tester to build and send a message through AF layer using source routing.

**Usage**:

**SREQ:**

| Byte: 1 | 1 | 1 | 2 | 1 |
|---|---|---|---|---|
| Length = 0x0B-0xFA | Cmd0 = 0x24 | Cmd1 = 0x03 | DstAddr | DstEndpoint |

| Byte: 1 | 2 | 1 | 1 | 1 | 1 | 2N | 1 | 0-128 |
|---|---|---|---|---|---|---|---|---|
| SrcEndpoint | ClusterId | TransId | Options | Radius | Relay Count (N) | RelayList | Len | Data |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| DstAddr | 2 | Short address of the destination device |
| DstEndpoint | 1 | Endpoint of the destination device |
| SrcEndpoint | 1 | Endpoint of the source device |
| ClusterId | 2 | Specifies the cluster ID |
| TransId | 1 | Specifies the transaction sequence number of the message. |
| Options | 1 | Transmit options bit mask: Bit 0: turns on/off 'APS ACK'; bit 2 sets 'APS security'; bit 3 sets 'skip routing'. |
| Radius | 1 | Specifies the number of hops allowed delivering the message (reference DEF_NWK_RADIUS.) |
| Relay Count | 1 | Specifies the number of devices in the relay list for source routing |
| Relay List | 2N | List of relay devices on the source routing path. For each device, it contains 2 bytes short address for each device. |
| Len | 1 | Length of the data. |
| Data | 0-128 | 0-128 bytes data |

**SRSP:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x64 | Cmd1 = 0x03 | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Memory Failure (0x01). |

### 3.2.1.5    AF_INTER_PAN_CTL

**Description**:

Inter-Pan control command and data. The data content depends upon the command and the available commands are enumerated as InterPanCtl_t.

**Usage**:

**SREQ:**

| Byte: 1 | 1 | 1 | 1 | 0-3 |
|---|---|---|---|---|
| Length = 0x01-0x04 | Cmd0 = 0x24 | Cmd1 = 0x10 | Command | Data |

**Data**:

| Command | Data Length (byte) | Description |
|---|---|---|
| 0: InterPanClr | 0 | Proxy call to StubAPS_SetIntraPanChannel() to switch channel back to the NIB-specified channel. |
| 1: InterPanSet | 1 | Proxy call to StubAPS_SetInterPanChannel() with the 1-byte channel specified. |
| 2: InterPanReg | 1 | If the 1-byte Endpoint specified by the data argument is found by invoking afFindEndPointDesc(), then proxy a call to StubAPS_RegisterApp() with the pointer to the endPointDesc_t found (i.e. the Endpoint must already be registered with AF). |
| 3: InterPanChk | 3 | Proxy a call to StubAPS_InterPan() with the 2-byte PanId (LSB:MSB) and 1-byte EndPoint data. |

**SRSP:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x64 | Cmd1 = 0x10 | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Success (0) <br><br> Failure (0x01) if a channel change is in progress <br><br> Invalid_Parameter (0x02). <br><br> ZApsNotAllowed (0xBA) if MAC is not in an Idle state. |

### 3.2.1.6    AF_ DATA_STORE

**Description**:

Huge AF data request data buffer store command and data.

**Usage**:

**SREQ:**

| Byte: 1 | 1 | 1 | 2 | 1 | 0-252 |
|---|---|---|---|---|---|
| Length = 0x03-0xFA | Cmd0 = 0x24 | Cmd1 = 0x11 | Index | Length | Data |

**Attributes**:

| Command | Length (byte) | Description |
|---|---|---|
| Index | 2 | Specifies the index into the outgoing data request data buffer to start the storing of this chunk of data. |
| Length | 1 | Specifies the length of this data chunk to store.<br><br>A length of zero is special and triggers the actually sending of the data request OTA. |
| Data | 0-252 | Contains 0 to 252 bytes of data. |

**SRSP:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x64 | Cmd1 = 0x11 | AF-Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | afStatus_SUCCESS          0x00<br><br>afStatus_FAILED           0x01<br><br>afStatus_MEM_FAIL         0x10<br><br>afStatus_INVALID_PARAMETER  0x02<br><br><br>Note that the status is for storing a chunk of data when Length is not zero and the return value of the AF_DataRequest() when it is zero. |

### 3.2.1.7    AF_ DATA_RETRIEVE

**Description**:

Huge AF incoming message data buffer retrieve command.

**Usage**:

**SREQ:**

| Byte: 1 | 1 | 1 | 4 | 2 | 1 |
|---|---|---|---|---|---|
| Length = 0x07 | Cmd0 = 0x24 | Cmd1 = 0x12 | Timestamp | Index | Length |

**Attributes**:

| Command | Length | Description |
|---|---|---|
| Timestamp | 4 | The timestamp of the incoming message in order to uniquely Identify it in a queue of incoming huge messages. |
| Index | 2 | Specifies the index into the incoming message data buffer to start the retrieving of this chunk of data. |
| Length | 1 | Specifies the length of this data chunk to retrieve. A length of zero is special and triggers the freeing of the corresponding incoming message. |

**SRSP:**

| Byte: 1 | 1 | 1 | 1 | 1 | 0-253 |
|---|---|---|---|---|---|
| Length = 0x02-0xFA | Cmd0 = 0x64 | Cmd1 = 0x12 | AF-Status | Length | Data |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | afStatus_SUCCESS         0x00 <br> afStatus_FAILED          0x01 <br> afStatus_MEM_FAIL         0x10 <br> afStatus_INVALID_PARAMETER  0x02 |
| Length | 1 | Specifies the length of this data chunk retrieved. |
| Data | 0-253 | The length of data bytes requested from the specified index into the huge incoming message data buffer. |

### 3.2.1.8    AF_APSF_CONFIG_SET

**Description**:

MT proxy for afAPSF_ConfigSet().

**Usage**:

**SREQ:**

| Byte: 1 | 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|---|
| Length = 0x03 | Cmd0 = 0x24 | Cmd1 = 0x13 | endPoint | frameDelay | windowSize |

**Attributes**:

| Command | Length | Description |
|---|---|---|
| endpoint | 1 | The specific EndPoint for which to set the fragmentation configuration. |
| frameDelay | 1 | The APS Fragmentation inter-frame delay in milliseconds (delay between frame fragments within the window). |
| windowSize | 1 | The APS Fragmentation window size (number of frames between APS acknowledge from receiving node). |

**SRSP:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x64 | Cmd1 = 0x13 | AF-Status |

**Attributes**:

| Attribute | Length (byte) | Description | |
|---|---|---|---|
| Status | 1 | afStatus_SUCCESS | 0x00 |
| | | afStatus_INVALID_PARAMETER | 0x02 |

### 3.2.2   MT_AF Callbacks

#### 3.2.1.1      AF_DATA_CONFIRM

**Description:**

This command is sent by the device to the user after it receives a data request.

**Usage:**

**AREQ:**

| Byte: 1 | 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|---|
| Length = 0x03 | Cmd0 = 0x44 | Cmd1 = 0x80 | Status | Endpoint | TransId |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |
| Endpoint | 1 | Endpoint of the device |
| TransId | 1 | Specified the transaction sequence number of the message |

#### 3.2.1.2      AF_REFLECT_ERROR

**Description:**

This command is sent by the device to the user when it determines that an error occurred during a reflected message.

**Usage:**

**AREQ:**

| Byte: 1 | 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|---|
| Length = 0x06 | Cmd0 = 0x44 | Cmd1 = 0x83 | Status | Endpoint | TransId |

| 1 | 2 |
|---|---|
| dstAddrMode | dstAddr |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Contains the error indication (ie. ZApsNoAck) |
| Endpoint | 1 | Endpoint of the device |
| TransId | 1 | Specified the transaction sequence number of the message |
| dstAddrMode | 1 | Destination address type: 0 - short address, 1 - group address |
| dstAddr | 2 | Destination address - depends on dstAddrMode |

### 3.2.1.3    AF_INCOMING_MSG

**Description:**

This callback message is in response to incoming data to any of the registered endpoints on this device.

**Usage:**

**AREQ:**

| 1 | 1 | 1 | 2 | 2 | 2 | 1 |
|---|---|---|---|---|---|---|
| Length = 0x11-0x91 | Cmd0 = 0x44 | Cmd1 = 0x81 | GroupId | ClusterId | SrcAddr | SrcEndpoint |

| 1 | 1 | 1 | 1 | 4 | 1 | 1 | 0-128 |
|---|---|---|---|---|---|---|---|
| DstEndpoint | WasBroadcast | LinkQuality | SecurityUse | Timestamp | TransSeqNumber | Len | Data |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| GroupId | 2 | Specifies the group ID of the device |
| ClusterId | 2 | Specifies the cluster Id (only the LSB is used in V1.0 networks.) |
| SrcAddr | 2 | Specifies the ZigBee network address of the source device sending the message. |
| SrcEndpoint | 1 | Specifies the source endpoint of the message |
| DstEndpoint | 1 | Specifies the destination endpoint of the message |
| WasBroadcast | 1 | Specifies if the message was a broadcast or not |
| LinkQuality | 1 | Indicates the link quality measured during reception |
| SecurityUse | 1 | Specifies if the security is used or not |
| TimeStamp | 4 | Specifies the timestamp of the message |
| TransSeqNumber | 1 | Specifies transaction sequence number of the message |
| Len | 1 | Specifies the length of the data. |
| Data | 0-128 | Contains 0 to 128 bytes of data. |

### 3.2.1.4    AF_INCOMING_MSG_EXT

**Description:**

This callback message is in response to incoming data to any of the registered endpoints on this device when the code is compiled with the INTER_PAN flag defined. This extended incoming message indication must also be used when handling an incoming message with a huge data byte count which is defined to be a size that would cause the RPC request to exceed the maximum allowed size:
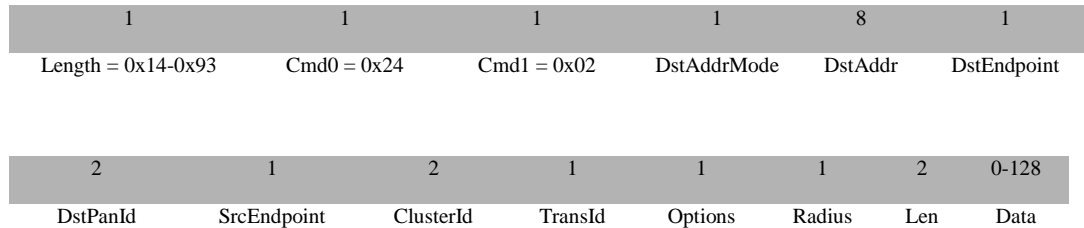
```
            MT_RPC_DATA_MAX – sizeof(AF_INCOMING_MSG_EXT)
```

Where `sizeof(AF_ INCOMING_MSG _EXT)` counts everything but the data bytes and now stands at 27. An AF_INCOMING_MSG_EXT with a huge data byte count indication shall not contain any data bytes. The huge data buffer must be retrieved separately as a sequence of one or more AF_DATA_RETRIEVE requests. Note that the incoming huge message is timed-out in 15 seconds after receiving it; thus all AF_DATA_ RETRIEVE requests must be completed within 15 seconds of an AF_ INCOMING_MSG _EXT with a huge data byte count. Note that multiple AF_INCOMING_MSG_EXT indications with huge data byte counts may be queued, and each will be timed-out separately. The default timeout can be changed by defining the following to other values:

```
#if !defined MT_AF_EXEC_CNT

#define MT_AF_EXEC_CNT  15

#endif

#if !defined MT_AF_EXEC_DLY

#define MT_AF_EXEC_DLY  1000

#endif
```

**Usage:**

**AREQ:**

| 1 | 1 | 1 | 2 | 2 | 1 | 8 | 1 |
|---|---|---|---|---|---|---|---|
| Length = 0x1B-0x9A | Cmd0 = 0x44 | Cmd1 = 0x82 | GroupId | ClusterId | SrcAddr Mode | SrcAddr | SrcEndpoint |

| 2 | 1 | 1 | 1 | 1 | 4 | 1 |
|---|---|---|---|---|---|---|
| SrcPanId | DstEndpoint | WasBroadcast | LinkQuality | SecurityUse | Timestamp | TransSeqNumber |

| 1 | 0-128 |
|---|---|
| Len | Data |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| GroupId | 2 | Specifies the group ID of the device |
| ClusterId | 2 | Specifies the cluster Id (only the LSB is used in V1.0 networks.) |
| SrcAddrMode | 1 | A value of 3 (i.e. the enumeration value for 'afAddr64Bit') indicates 8-byte/64-bit address mode; otherwise, only the 2 LSB's of the 8 bytes are used to form a 2-byte short address. |
| SrcAddr | 8 | LSB to MSB for the long or short address of the destination device (upper 6 bytes are don't care when short address.) |
| SrcEndpoint | 1 | Specifies the source endpoint of the message |
| SrcPanId | 2 | Specifies the source PanID of the message. |
| DstEndpoint | 1 | Specifies the destination endpoint of the message |

| | | |
|---|---|---|
| WasBroadcast | 1 | Specifies if the message was a broadcast or not |
| LinkQuality | 1 | Indicates the link quality measured during reception |
| SecurityUse | 1 | Specifies if the security is used or not |
| TimeStamp | 4 | Specifies the timestamp of the message |
| TransSeqNumber | 1 | Specifies transaction sequence number of the message |
| Len | 2 | Specifies the length of the data. If a large data length causes the MT command to exceed MT_RPC_DATA_MAX, then zero bytes of the data shall be sent with this request and the host shall retrieve the data with as many MT_AF_DATA_RETRIEVE requests as necessary. |
| Data | 0-128 | Contains 0 to 128 bytes of data. |

## 3.3    MT_APP

This interface allows tester to interact with APP layer of the target to control custom tests such as test profile or user-defined test.

### 3.3.1   MT_APP Commands

#### 3.3.1.1     APP_MSG

**Description:**

This command is sent to the target in order to test the functions defined for individual applications. This command sends a raw data to an application.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 1 | 2 |
|---|---|---|---|---|
| Length = 0x07-0x87 | Cmd0 = 0x29 | Cmd1 = 0x00 | AppEndpoint | DestAddress |

| 1 | 2 | 1 | 0-128 |
|---|---|---|---|
| DestEndpoint | ClusterId | MsgLen | Message |

**Attributes**:

| Attribute | Length (byte) | Description |
|-----------|---------------|-------------|
| AppEndpoint | 1 | Application endpoint of the outgoing message |
| DestAddress | 2 | Destination address of the outgoing message |
| DestEndpoint | 1 | Destination endpoint of the outgoing message |
| ClusterId | 2 | Cluster Id of the outgoing message |
| MsgLen | 1 | Length of the outgoing message |
| Message | 0-128 | Raw data packet to send to the application |

**SRSP**:

| Byte: 1 | 1 | 1 | 1 |
|---------|---|---|---|
| Length = 0x01 | Cmd0 = 0x69 | Cmd1 = 0x00 | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|-----------|---------------|-------------|
| Status | 1 | Status is either Success (0) or Failure (1). |

## 3.3.1.2    APP_USER_TEST

**Description:**

This command is used by tester to issue user's defined commands to the application.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 1 | 2 | 2 | 2 |
|---------|---|---|---|---|---|---|
| Length = 0x07 | Cmd0 = 0x29 | Cmd1 = 0x01 | SrcEP | CommandId | Parameter1 | Parameter2 |

**Attributes**:

| Attribute | Length (byte) | Description |
|-----------|---------------|-------------|
| SrcEP | 1 | Source Endpoint of the user-defined command |
| CommandId | 2 | Command Id of the user-defined command |
| Parameter1 | 2 | Parameter #1 of the command |
| Parameter2 | 2 | Parameter #2 of the command |

**SRSP**:

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x69 | Cmd1 = 0x01 | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

## 3.3.2  MT_APP Callbacks

NONE

## 3.4     MT_DEBUG

This interface allows tester to control the debug-messaging mechanism such as debug threshold, debug messages, etc.

## 3.4.1  MT_DEBUG Commands

### 3.4.1.1     DEBUG_SET_THRESHOLD

**Description**:

This command allows the user to set the threshold for the debug message.

**Usage**:

**SREQ:**

| Byte: 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|
| Length = 0x03 | Cmd0 = 0x28 | Cmd1 = 0x00 | ComponentId | Threshold |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| ComponentId | 1 | Uniquely Identifies a particular software component on the target |
| Threshold | 1 | Specifies the threshold value for reporting debug messages by that software component |

**SRSP**:

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|

| Length = 0x01 | Cmd0 = 0x68 | Cmd1 = 0x00 | Status |
|---|---|---|---|

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.4.1.2    DEBUG_MSG

**Description:**

This command sends a debug string to Z-Tool. The content of the string is defined by the application.

**Usage:**

**AREQ:**

| Byte: 1 | 1 | 1 | 1 | 0-254 |
|---|---|---|---|---|
| Length = 0x01-0xFA | Cmd0 = 0x48 | Cmd1 = 0x00 | Length | String |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Length | 1 | Length of the string |
| String | 0-254 | String to be displayed by Z-Tool 2.0 |

## 3.4.2  MT_DEBUG Callbacks

NONE

## 3.5    MT_MAC

This interface allows tester to interact with the TI-MAC

### 3.5.1    MT_MAC Commands

#### 3.5.1.1      MAC_RESET_REQ

**Description:**

This command is used to send a MAC Reset command to reset MAC state machine.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x02 | Cmd0 = 0x22 | Cmd1 = 0x01 | SetDefault |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| SetDefault | 1 | TRUE – Set the MAC PIB values to default values. |

**SRSP**:

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x62 | Cmd1 = 0x00 | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.5.1.2    MAC_INIT

**Description:**

This command is used to initialize the MAC.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 |
|---|---|---|
| Length = 0x00 | Cmd0 = 0x22 | Cmd1 = 0x02 |

**Attributes**:

None

**SRSP**:

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x62 | Cmd1 = 0x02 | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.5.1.3    MAC_START_REQ

**Description:**

This command is used to start the MAC as a coordinator or end device.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 4 | 2 | 1 | 1 |
|---|---|---|---|---|---|---|
| Length = 0x1C | Cmd0 = 0x22 | Cmd1 = 0x03 | StartTime | PanId | LogicalChannel | ChannelPage |

| Byte: 1 | 1 | 1 | 1 | 1 | 8 |
|---|---|---|---|---|---|
| BeaconOrder | SuperFrameOrder | PanCoordinator | BatteryLifeExt | CoordRealignment | RealignKeySource |

| Byte: 1 | 1 | 1 | 8 | 1 |
|---|---|---|---|---|
| RealignSecurityLevel | RealignKeyIdMode | RealignKeyIndex | BeaconKeySource | BeaconSecurityLevel |

| Byte: 1 | 1 |
|---|---|
| BeaconKeyIdMode | BeaconKeyIndex |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| StartTime | 4 | The time to begin transmitting beacons relative to the received beacon |
| PanId | 2 | The PAN Id to use.  This parameter is ignored if Pan Coordinator is FALSE |
| LogicalChannel | 1 | The logical channel to use.  This parameter is ignored if Pan Coordinator is FALSE |
| ChannelPage | 1 | The channel page to use.  This parameter is ignored if Pan Coordinator is FALSE |
| BeaconOrder | 1 | The exponent used to calculate the beacon interval |
| SuperFrameOrder | 1 | The exponent used to calculate the superframe duration |
| PanCoordinator | 1 | Set to TRUE to start a network as PAN coordinator |
| BatteryLifeExt | 1 | If this value is TRUE, the receiver is disabled after MAC_BATT_LIFE_EXT_PERIODS full backoff periods following the interframe spacing period of the beacon frame |
| CoordRealignment | 1 | Coordinator Realignment |
| RealignKeySource | 8 | Key Source of this data frame |
| RealignSecurityLevel | 1 | Security Level of this data frame: |

| Security Level | Value |
|---|---|
| NO_SECURITY | 0x00 |
| MIC_32_AUTH | 0x01 |
| MIC_64_AUTH | 0x02 |
| MIC_128_AUTH | 0x03 |
| AES_ENCRYPTION | 0x04 |
| AES_ENCRYPTION_MIC_32 | 0x05 |
| AES_ENCRYPTION_MIC_64 | 0x06 |
| AES_ENCRYPTION_MIC_128 | 0x07 |

Key Id Mode of this data frame:

| Attribute | Length (byte) | Description |
|---|---|---|
| RealignKeyIdMode | 1 | |

| Key Id Mode | Value |
|---|---|
| NOT_USED | 0x00 |
| KEY_1BYTE_INDEX | 0x01 |
| KEY_4BYTE_INDEX | 0x02 |
| KEY_8BYTE_INDEX | 0x03 |

| Attribute | Length (byte) | Description |
|---|---|---|
| RealignKeyIndex | 1 | Key Index of this data frame |
| BeaconKeySource | 8 | Key Source of this data frame |

Security Level of this data frame:

| Security Level | Value |
|---|---|
| NO_SECURITY | 0x00 |
| MIC_32_AUTH | 0x01 |
| MIC_64_AUTH | 0x02 |
| MIC_128_AUTH | 0x03 |
| AES_ENCRYPTION | 0x04 |
| AES_ENCRYPTION_MIC_32 | 0x05 |
| AES_ENCRYPTION_MIC_64 | 0x06 |
| AES_ENCRYPTION_MIC_128 | 0x07 |

BeaconSecurityLevel          1

BeaconKeyIdMode          1          Key Id Mode of this data frame

BeaconKeyIndex          1          Key Index of this data frame

**SRSP**:

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x62 | Cmd1 = 0x03 | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

## 3.5.1.4    MAC_SYNC_REQ

**Description:**

This command is used to request synchronization to the current network beacon

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|---|
| Length = 0x03 | Cmd0 = 0x22 | Cmd1 = 0x04 | LogicalChannel | ChannelPage | TrackBeacon |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| LogicalChannel | 1 | The logical channel to use. |
| ChannelPage | 1 | The channel page to use. |
| TrackBeacon | 1 | Set to TRUE to continue tracking beacons after synchronizing with the first beacon.  Set to FALSE to only synchronize with the first beacon |

**SRSP**:

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x62 | Cmd1 = 0x04 | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.5.1.5    MAC_DATA_REQ

**Description:**

This command is used to send (on behalf of the next higher layer) MAC Data Frame packet.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 1 | 8 | 2 |
|---|---|---|---|---|---|
| Length = 0x15-0x114 | Cmd0 = 0x22 | Cmd1 = 0x05 | DestAddressMode | DestAddress | DestPanId |

| Byte: 1 | 1 | 1 | 1 | 1 | 8 | 1 | 1 |
|---|---|---|---|---|---|---|---|
| SrcAddressMode | Handle | TxOption | LogicalChannel | Power | KeySource | SecurityLevel | KeyIdMode |

| Byte: 1 | 1 | 0-250 |
|---|---|---|
| KeyIndex | MSDULength | MSDU |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| | | Specifies the format of the destination address. |
| DestAddressMode | 1 | <table><tr><td>Mode</td><td>Value</td><td>Description</td></tr><tr><td>ADDRESS_NOT_PRESENT</td><td>0x00</td><td>Address Not Present</td></tr><tr><td>GROUP_ADDRESS</td><td>0x01</td><td>Group address</td></tr><tr><td>ADDRESS_16_BIT</td><td>0x02</td><td>Address 16 bit</td></tr><tr><td>ADDRESS_64_BIT</td><td>0x03</td><td>Address 64 bit</td></tr><tr><td>BROADCAST</td><td>0xFF</td><td>Broadcast</td></tr></table> |
| DestAddress | 8 | Address of the destination. |
| DestPanId | 2 | PAN Id of the destination. |
| | | Specifies the format of the source address. |
| SrcAddressMode | 1 | <table><tr><td>Mode</td><td>Value</td><td>Description</td></tr><tr><td>ADDRESS_NOT_PRESENT</td><td>0x00</td><td>Address Not Present</td></tr><tr><td>GROUP_ADDRESS</td><td>0x01</td><td>Group address</td></tr><tr><td>ADDRESS_16_BIT</td><td>0x02</td><td>Address 16 bit</td></tr><tr><td>ADDRESS_64_BIT</td><td>0x03</td><td>Address 64 bit</td></tr><tr><td>BROADCAST</td><td>0xFF</td><td>Broadcast</td></tr></table> |
| Handle | 1 | Handle of the packet. |

Transmitting options:

| Option | Value | Description |
|---|---|---|
| MAC_TXOPTION_ACK | 0x01 | Acknowledged transmission. The MAC will attempt to retransmit the frame until it is acknowledged |
| MAC_TXOPTION_GTS | 0x02 | GTS transmission (unused) |
| MAC_TXOPTION_INDIRECT | 0x04 | Indirect transmission.  The MAC will queue the data and wait for the destination device to poll for it.  This can only be used by a coordinator device |
| MAC_TXOPTION_NO_RETRANS | 0x10 | This proprietary option prevents the frame from being retransmitted |
| MAC_TXOPTION_NO_CNF | 0x20 | This proprietary option prevents a MAC_MCPS_DATA_CNF event from being sent for this frame |
| MAC_TXOPTION_ALT_BE | 0x40 | Use PIB value MAC_ALT_BE for the minimum backoff exponent |
| MAC_TXOPTION_PWR_CHAN | 0x80 | Use the power and channel values in macDataReq_t instead of the PIB values |

TxOption              1

LogicalChannel        1        Channel that data frame will be transmitted.

Power                 1        Power level that data frame will be transmitted.

KeySource             8        Key Source of this data frame.

Security Level of this data frame:

SecurityLevel         1

| Security Level | Value |
|---|---|
| NO_SECURITY | 0x00 |
| MIC_32_AUTH | 0x01 |
| MIC_64_AUTH | 0x02 |
| MIC_128_AUTH | 0x03 |
| AES_ENCRYPTION | 0x04 |
| AES_ENCRYPTION_MIC_32 | 0x05 |
| AES_ENCRYPTION_MIC_64 | 0x06 |
| AES_ENCRYPTION_MIC_128 | 0x07 |

Key Id Mode of this data frame:

| Key Id Mode | Value |
|---|---|
| NOT_USED | 0x00 |
| KEY_1BYTE_INDEX | 0x01 |
| KEY_4BYTE_INDEX | 0x02 |
| KEY_8BYTE_INDEX | 0x03 |

| | | |
|---|---|---|
| KeyIdMode | 1 | (see table above) |
| KeyIndex | 1 | Key Index of this data frame. |
| MSDULength | 1 | Length of the data. |
| MSDU | 0-250 | Actual data that will be sent. |

**SRSP**:

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x62 | Cmd1 = 0x05 | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

## 3.5.1.6    MAC_ASSOCIATE_REQ

**Description:**

This command is used to request (on behalf of the next higher layer) an association with a coordinator

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|---|
| Length = 0x12 | Cmd0 = 0x22 | Cmd1 = 0x06 | LogicalChannel | ChannelPage | CoordAddressMode |

| Byte: 8 | 2 | 1 | 8 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|
| CoordAddress | CoordPanId | CapabilityInformation | KeySource | SecurityLevel | KeyIdMode | KeyIndex |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| LogicalChannel | 1 | Channel that data frame will be transmitted. |
| ChannelPage | 1 | The channel page to be used. |
| CoordAddressMode | 1 | Specifies the format of the coordinator address. |

| Mode | Value | Description |
|---|---|---|
| ADDRESS_NOT_PRESENT | 0x00 | Address Not Present |
| GROUP_ADDRESS | 0x01 | Group address |
| ADDRESS_16_BIT | 0x02 | Address 16 bit |
| ADDRESS_64_BIT | 0x03 | Address 64 bit |
| BROADCAST | 0xFF | Broadcast |

| Attribute | Length (byte) | Description |
|---|---|---|
| CoordAddress | 8 | Address of the Coordinator. |
| CoordPanId | 2 | PAN Id of the coordinator. |
| CapabilityInformation | 1 | Bit map which specifies the operational capabilities of the device.<br><br>Bit: 0 – Alternate PAN Coordinator<br><br>1 – Device type: 1- ZigBee Router; 0 – End Device<br><br>2 – Power Source: 1 Main powered<br><br>3 – Receiver on when Idle<br><br>4 – Reserved<br><br>5 – Reserved<br><br>6 – Security capability<br><br>7 – Reserved |
| KeySource | 8 | Key Source of this data frame |
| SecurityLevel | 1 | Security Level of this data frame: |

| Security Level | Value |
|---|---|
| NO_SECURITY | 0x00 |
| MIC_32_AUTH | 0x01 |
| MIC_64_AUTH | 0x02 |
| MIC_128_AUTH | 0x03 |
| AES_ENCRYPTION | 0x04 |

| AES_ENCRYPTION_MIC_32 | 0x05 |
|---|---|
| AES_ENCRYPTION_MIC_64 | 0x06 |
| AES_ENCRYPTION_MIC_128 | 0x07 |

Key Id Mode of this data frame:

| | | Key Id Mode | Value |
|---|---|---|---|
| KeyIdMode | 1 | NOT_USED | 0x00 |
| | | KEY_1BYTE_INDEX | 0x01 |
| | | KEY_4BYTE_INDEX | 0x02 |
| | | KEY_8BYTE_INDEX | 0x03 |
| KeyIndex | 1 | Key Index of this data frame. | |

**SRSP**:

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x62 | Cmd1 = 0x06 | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.5.1.7    MAC_ASSOCIATE_RSP

**Description:**

This command is sent by the host to response to the MAC_ASSOCIATE_IND.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 8 | 2 | 1 |
|---|---|---|---|---|---|
| Length = 0x0B | Cmd0 = 0x42 | Cmd1 = 0x50 | ExtAddr | AssocShortAddress | AssocStatus |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| ExtAddr | 8 | Extended Address of the device requesting association |
| AssocShortAddress | 2 | Short address for the associated device. Allocated by the coordinator. |
| AssocStatus | 1 | Status of the association:<br><br>| Status | Value |<br>|---|---|<br>| SUCCESSFUL_ASSOCIATION | 0x00 |<br>| PAN_AT_CAPACITY | 0x01 |<br>| PAN_ACCESS_DENIED | 0x02 | |

**SRSP**:

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x62 | Cmd1 = 0x50 | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.5.1.8    MAC_DISASSOCIATE_REQ

**Description:**

This command is used to request (on behalf of the next higher layer) a disassociation of the device from the coordinator.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 1 | 8 | 2 |
|---|---|---|---|---|---|
| Length = 0x18 | Cmd0 = 0x22 | Cmd1 = 0x07 | DeviceAddressMode | DeviceAddress | DevicePanId |

| Byte: 1 | 1 | 8 | 1 | 1 | 1 |
|---|---|---|---|---|---|
| DisassociateReason | TxIndirect | KeySource | SecurityLevel | KeyIdMode | KeyIndex |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| DeviceAddressMode | 1 | Specifies the format of the device address.<br><br>| Mode | Value | Description |<br>|---|---|---|<br>| ADDRESS_NOT_PRESENT | 0x00 | Address Not Present |<br>| GROUP_ADDRESS | 0x01 | Group address |<br>| ADDRESS_16_BIT | 0x02 | Address 16 bit |<br>| ADDRESS_64_BIT | 0x03 | Address 64 bit |<br>| BROADCAST | 0xFF | Broadcast | |
| DeviceAddress | 8 | Device Address. |
| DevicePanId | 2 | Network PAN Id of device. |
| DisassociateReason | 1 | Reason of disassociation:<br><br>| Reason | Value |<br>|---|---|<br>| RESERVED | 0x00 |<br>| COOR_WISHES_DEV_LEAVE | 0x01 |<br>| DEV_WISHES_LEAVE | 0x02 | |
| TxIndirect | 1 | Tx indirect |
| KeySource | 8 | Key Source of this data frame. |

Security Level of this data frame:

| Security Level | Value |
|---|---|
| NO_SECURITY | 0x00 |
| MIC_32_AUTH | 0x01 |
| MIC_64_AUTH | 0x02 |
| MIC_128_AUTH | 0x03 |
| AES_ENCRYPTION | 0x04 |
| AES_ENCRYPTION_MIC_32 | 0x05 |
| AES_ENCRYPTION_MIC_64 | 0x06 |
| AES_ENCRYPTION_MIC_128 | 0x07 |

SecurityLevel          1

Key Id Mode of this data frame:

| Key Id Mode | Value |
|---|---|
| NOT_USED | 0x00 |
| KEY_1BYTE_INDEX | 0x01 |
| KEY_4BYTE_INDEX | 0x02 |
| KEY_8BYTE_INDEX | 0x03 |

KeyIdMode          1

KeyIndex          1          Key Index of this data frame.

**SRSP**:

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x62 | Cmd1 = 0x07 | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.5.1.9    MAC_GET_REQ

**Description:**

This command is used to read (on behalf of the next higher layer) a MAC PIB attribute.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x22 | Cmd1 = 0x08 | Attribute |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| | | Specifies the MAC PIB Attributes: |

| MAC PIB Attribute | Value |
|---|---|
| ZMAC_ACK_WAIT_DURATION | 0x40 |
| ZMAC_ASSOCIATION_PERMIT | 0x41 |
| ZMAC_AUTO_REQUEST | 0x42 |
| ZMAC_BATT_LIFE_EXT | 0x43 |
| ZMAC_BATT_LEFT_EXT_PERIODS | 0x44 |
| ZMAC_BEACON_MSDU | 0x45 |
| ZMAC_BEACON_MSDU_LENGTH | 0x46 |
| ZMAC_BEACON_ORDER | 0x47 |
| ZMAC_BEACON_TX_TIME | 0x48 |
| ZMAC_BSN | 0x49 |
| ZMAC_COORD_EXTENDED_ADDRESS | 0x4A |
| ZMAC_COORD_SHORT_ADDRESS | 0x4B |
| ZMAC_DSN | 0x4C |
| ZMAC_GTS_PERMIT | 0x4D |
| ZMAC_MAX_CSMA_BACKOFFS | 0x4E |
| ZMAC_MIN_BE | 0x4F |
| ZMAC_PANID | 0x50 |
| ZMAC_PROMISCUOUS_MODE | 0x51 |

(Attribute row label: "Attribute    1" aligned with ZMAC_BEACON_ORDER)

| ZMAC_RX_ON_IDLE | 0x52 |
|---|---|
| ZMAC_SHORT_ADDRESS | 0x53 |
| ZMAC_SUPERFRAME_ORDER | 0x54 |
| ZMAC_TRANSACTION_PERSISTENCE_TIME | 0x55 |
| ZMAC_ASSOCIATED_PAN_COORD | 0x56 |
| ZMAC_MAX_BE | 0x57 |
| ZMAC_FRAME_TOTAL_WAIT_TIME | 0x58 |
| ZMAC__MAC_FRAME_RETRIES | 0x59 |
| ZMAC_RESPONSE_WAIT_TIME | 0x5A |
| ZMAC_SYNC_SYMBOL_OFFSET | 0x5B |
| ZMAC_TIMESTAMP_SUPPORTED | 0x5C |
| ZMAC_SECURITY_ENABLED | 0x5D |
| ZMAC_PHY_TRANSMIT_POWER | 0xE0 |
| ZMAC_LOGICAL_CHANNEL | 0xE1 |
| ZMAC_EXTENDED_ADDRESS | 0xE2 |
| ZMAC_ALT_BE | 0xE3 |

**SRSP**:

| Byte: 1 | 1 | 1 | 1 | 16 |
|---|---|---|---|---|
| Length = 0x11 | Cmd0 = 0x62 | Cmd1 = 0x08 | Status | Data |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |
| Data | 16 | 1-16 bytes value of the PIB attribute. |

### 3.5.1.10    MAC_SET_REQ

**Description:**

This command is used to request the device to write a MAC PIB value.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 1 | 16 |
|---|---|---|---|---|
| Length = 0x11 | Cmd0 = 0x22 | Cmd1 = 0x09 | Attribute | AttributeValue |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| | | Specified the MAC PIB Attribute: |

| Attribute | 1 | MAC PIB Attribute | Value |
|---|---|---|---|
| | | ZMAC_ACK_WAIT_DURATION | 0x40 |
| | | ZMAC_ASSOCIATION_PERMIT | 0x41 |
| | | ZMAC_AUTO_REQUEST | 0x42 |
| | | ZMAC_BATT_LIFE_EXT | 0x43 |
| | | ZMAC_BATT_LEFT_EXT_PERIODS | 0x44 |
| | | ZMAC_BEACON_MSDU | 0x45 |
| | | ZMAC_BEACON_MSDU_LENGTH | 0x46 |
| | | ZMAC_BEACON_ORDER | 0x47 |
| | | ZMAC_BEACON_TX_TIME | 0x48 |
| | | ZMAC_BSN | 0x49 |
| | | ZMAC_COORD_EXTENDED_ADDRESS | 0x4A |
| | | ZMAC_COORD_SHORT_ADDRESS | 0x4B |
| | | ZMAC_DSN | 0x4C |
| | | ZMAC_GTS_PERMIT | 0x4D |
| | | ZMAC_MAX_CSMA_BACKOFFS | 0x4E |
| | | ZMAC_MIN_BE | 0x4F |
| | | ZMAC_PANID | 0x50 |
| | | ZMAC_PROMISCUOUS_MODE | 0x51 |

| | |
|---|---|
| ZMAC_RX_ON_IDLE | 0x52 |
| ZMAC_SHORT_ADDRESS | 0x53 |
| ZMAC_SUPERFRAME_ORDER | 0x54 |
| ZMAC_TRANSACTION_PERSISTENCE_TIME | 0x55 |
| ZMAC_ASSOCIATED_PAN_COORD | 0x56 |
| ZMAC_MAX_BE | 0x57 |
| ZMAC_FRAME_TOTAL_WAIT_TIME | 0x58 |
| ZMAC__MAC_FRAME_RETRIES | 0x59 |
| ZMAC_RESPONSE_WAIT_TIME | 0x5A |
| ZMAC_SYNC_SYMBOL_OFFSET | 0x5B |
| ZMAC_TIMESTAMP_SUPPORTED | 0x5C |
| ZMAC_SECURITY_ENABLED | 0x5D |
| ZMAC_PHY_TRANSMIT_POWER | 0xE0 |
| ZMAC_LOGICAL_CHANNEL | 0xE1 |
| ZMAC_EXTENDED_ADDRESS | 0xE2 |
| ZMAC_ALT_BE | 0xE3 |

AttributeValue        16        1-16 bytes of the PIB attribute value.

**SRSP**:

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x62 | Cmd1 = 0x09 | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.5.1.11    MAC_SCAN_REQ

**Description:**

This command is used to send a request to the device to perform a network scan.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 4 | 1 | 1 |
|---|---|---|---|---|---|
| Length = 0x13 | Cmd0 = 0x22 | Cmd1 = 0x0C | ScanChannels | ScanType | ScanDuration |

| Byte: 1 | 1 | 8 | 1 | 1 | 1 |
|---|---|---|---|---|---|
| ChannelPage | MaxResults | KeySource | SecurityLevel | KeyIdMode | KeyIndex |

**Attributes**:

| Attribute | Length (byte) | Description | |
|---|---|---|---|
| ScanChannels | 4 | | This represents a bit-mask of channels to be scanned when starting the device: |

| Channel | Value |
|---|---|
| NONE | 0x00000000 |
| ALL_CHANNELS | 0x07FFF800 |
| CHANNEL 11 | 0x00000800 |
| CHANNEL 12 | 0x00001000 |
| CHANNEL 13 | 0x00002000 |
| CHANNEL 14 | 0x00004000 |
| CHANNEL 15 | 0x00008000 |
| CHANNEL 16 | 0x00010000 |
| CHANNEL 17 | 0x00020000 |
| CHANNEL 18 | 0x00040000 |
| CHANNEL 19 | 0x00080000 |
| CHANNEL 20 | 0x00100000 |
| CHANNEL 21 | 0x00200000 |
| CHANNEL 22 | 0x00400000 |
| CHANNEL 23 | 0x00800000 |
| CHANNEL 24 | 0x01000000 |
| CHANNEL 25 | 0x02000000 |

| CHANNEL 26 | 0x04000000 |
|---|---|

| | | Scan Type | Value |
|---|---|---|---|
| ScanType | 1 | ENERGY_DETECT | 0x00 |
| | | ACTIVE | 0x01 |
| | | PASSIVE | 0x02 |
| | | ORPHAN | 0x03 |

| | | |
|---|---|---|
| ScanDuration | 1 | Duration of the scan - The exponent used in the scan duration calculation. |
| ChannelPage | 1 | The channel page on which to perform the scan. |
| KeySource | 8 | Key Source of this data frame. |

Security Level of this data frame:

| | | Security Level | Value |
|---|---|---|---|
| | | NO_SECURITY | 0x00 |
| | | MIC_32_AUTH | 0x01 |
| SecurityLevel | 1 | MIC_64_AUTH | 0x02 |
| | | MIC_128_AUTH | 0x03 |
| | | AES_ENCRYPTION | 0x04 |
| | | AES_ENCRYPTION_MIC_32 | 0x05 |
| | | AES_ENCRYPTION_MIC_64 | 0x06 |
| | | AES_ENCRYPTION_MIC_128 | 0x07 |

Key Id Mode of this data frame:

| | | Key Id Mode | Value |
|---|---|---|---|
| KeyIdMode | 1 | NOT_USED | 0x00 |
| | | KEY_1BYTE_INDEX | 0x01 |
| | | KEY_4BYTE_INDEX | 0x02 |
| | | KEY_8BYTE_INDEX | 0x03 |
| KeyIndex | 1 | Key Index of this data frame. | |

**SRSP**:

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x62 | Cmd1 = 0x0C | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.5.1.12   MAC_ORPHAN_RSP

**Description:**

This command is sent by the host to response to the ORPHAN_IND.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 8 | 2 | 1 |
|---|---|---|---|---|---|
| Length = 0x0B | Cmd0 = 0x42 | Cmd1 = 0x51 | ExtAddr | AssocShortAddress | AssociatedMember |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| ExtAddr | 8 | Extended Address of the device sending the orphan notification |
| AssocShortAddress | 2 | Short address of the orphan device |
| AssociatedMember | 1 | TRUE if the orphan is an associated member. FALSE otherwise. |

**SRSP**:

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x62 | Cmd1 = 0x51 | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

## 3.5.1.13   MAC_POLL_REQ

**Description:**

This command is used to send a MAC data request poll.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 1 | 8 | 2 |
|---|---|---|---|---|---|
| Length = 0x16 | Cmd0 = 0x22 | Cmd1 = 0x0D | CoordAddressMode | CoordAddress | CoordPanId |

| 8 | 1 | 1 | 1 |
|---|---|---|---|
| KeySource | SecurityLevel | KeyIdMode | KeyIndex |

**Attributes**:

| Attribute | Length (byte) | Description | | |
|---|---|---|---|---|
| | | Mode | Value | Description |
| | | ADDRESS_NOT_PRESENT | 0x00 | Address Not Present |
| CoordAddressMode | 1 | GROUP_ADDRESS | 0x01 | Group address |
| | | ADDRESS_16_BIT | 0x02 | Address 16 bit |
| | | ADDRESS_64_BIT | 0x03 | Address 64 bit |

| BROADCAST | 0xFF | Broadcast |
|---|---|---|

| CoordAddress | 8 | 64-bit Coordinator Address |
|---|---|---|
| CoordPanId | 2 | Coordinator PanId |
| KeySource | 8 | Key Source of this data frame. |

Security Level of this data frame:

| Security Level | Value |
|---|---|
| NO_SECURITY | 0x00 |
| MIC_32_AUTH | 0x01 |
| MIC_64_AUTH | 0x02 |
| MIC_128_AUTH | 0x03 |
| AES_ENCRYPTION | 0x04 |
| AES_ENCRYPTION_MIC_32 | 0x05 |
| AES_ENCRYPTION_MIC_64 | 0x06 |
| AES_ENCRYPTION_MIC_128 | 0x07 |

SecurityLevel — 1

Key Id Mode of this data frame:

| Key Id Mode | Value |
|---|---|
| NOT_USED | 0x00 |
| KEY_1BYTE_INDEX | 0x01 |
| KEY_4BYTE_INDEX | 0x02 |
| KEY_8BYTE_INDEX | 0x03 |

KeyIdMode — 1

| KeyIndex | 1 | Key Index of this data frame. |
|---|---|---|

**SRSP**:

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x62 | Cmd1 = 0x0D | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.5.1.14    MAC_PURGE_REQ

**Description:**

This command is used to send a request to the device to purge a data frame

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x22 | Cmd1 = 0x0E | MsduHandle |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| MsduHandle | 1 | Msdu Handle |

**SRSP**:

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x62 | Cmd1 = 0x0E | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.5.1.15    MAC_SET_RX_GAIN_REQ

**Description:**

This command is used to send a request to the device to set Rx gain.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x22 | Cmd1 = 0x0F | Mode |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Mode | 1 | PA/PNA mode – TRUE/FALSE |

**SRSP**:

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x62 | Cmd1 = 0x0F | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.5.2    MT_MAC Callbacks

### 3.5.2.1    MAC_SYNC_LOSS_IND

**Description:**

This callback is called by the MAC to send (on behalf of the next higher layer) an indication of the synchronization loss.

**Usage:**

**AREQ:**

| 1 | 1 | 1 | 1 | 2 | 1 | 1 |
|---|---|---|---|---|---|---|
| Length = 0x10 | Cmd0 = 0x42 | Cmd1 = 0x80 | Status | PanId | LogicalChannel | ChannelPage |

| 8 | 1 | 1 | 1 |
|---|---|---|---|
| KeySource | SecurityLevel | KeyIdMode | KeyIndex |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| | | |

| Status | 1 | This field indicates either SUCCESS (0) or FAILURE (1). |
| PanId | 2 | PAN Id of the device |
| LogicalChannel | 1 | Logical Channel of the device where the synchronization is lost |
| ChannelPage | 1 | Channel Page of the device where the synchronization is lost |
| KeySource | 8 | Key Source of this data frame. |

Security Level of this data frame:

| Security Level | Value |
|---|---|
| NO_SECURITY | 0x00 |
| MIC_32_AUTH | 0x01 |
| MIC_64_AUTH | 0x02 |
| MIC_128_AUTH | 0x03 |
| AES_ENCRYPTION | 0x04 |
| AES_ENCRYPTION_MIC_32 | 0x05 |
| AES_ENCRYPTION_MIC_64 | 0x06 |
| AES_ENCRYPTION_MIC_128 | 0x07 |

SecurityLevel — 1

Key Id Mode of this data frame:

| Key Id Mode | Value |
|---|---|
| NOT_USED | 0x00 |
| KEY_1BYTE_INDEX | 0x01 |
| KEY_4BYTE_INDEX | 0x02 |
| KEY_8BYTE_INDEX | 0x03 |

KeyIdMode — 1

| KeyIndex | 1 | Key Index of this data frame. |

## 3.5.2.2    MAC_ASSOCIATE_IND

**Description:**

This callback is called by the MAC to send (on behalf of the next higher layer) an association indication message.

**Usage:**

**AREQ:**

| 1 | 1 | 1 | 8 | 1 |
|---|---|---|---|---|
| Length = 0x14 | Cmd0 = 0x42 | Cmd1 = 0x81 | DeviceExtendedAddress | Capabilities |

| 8 | 1 | 1 | 1 |
|---|---|---|---|
| KeySource | SecurityLevel | KeyIdMode | KeyIndex |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| DeviceExtendedAddress | 8 | Extended address of the device |
| Capabilities | 1 | Specifies the operating capabilities of the device being directly joined. Bit weighted values follow:<br><br>Bit: 0 – Alternate PAN Coordinator<br><br>1 – Device type: 1- ZigBee Router; 0 – End Device<br><br>2 – Power Source: 1 Main powered<br><br>3 – Receiver on when Idle<br><br>4 – Reserved<br><br>5 – Reserved<br><br>6 – Security capability<br><br>7 – Reserved |
| KeySource | 8 | Key Source of this data frame. |
| SecurityLevel | 1 | Security Level of this data frame: |

| Security Level | Value |
|---|---|
| NO_SECURITY | 0x00 |
| MIC_32_AUTH | 0x01 |
| MIC_64_AUTH | 0x02 |
| MIC_128_AUTH | 0x03 |
| AES_ENCRYPTION | 0x04 |
| AES_ENCRYPTION_MIC_32 | 0x05 |
| AES_ENCRYPTION_MIC_64 | 0x06 |
| AES_ENCRYPTION_MIC_128 | 0x07 |

Key Id Mode of this data frame:

| Key Id Mode | Value |
|---|---|
| NOT_USED | 0x00 |
| KEY_1BYTE_INDEX | 0x01 |
| KEY_4BYTE_INDEX | 0x02 |
| KEY_8BYTE_INDEX | 0x03 |

| KeyIdMode | 1 | |

KeyIndex                    1                Key Index of this data frame.

### 3.5.2.3    MAC_ASSOCIATE_CNF

**Description:**

This callback is called by the MAC to send (on behalf of the next higher layer) an association confirmation message.

**Usage:**

**AREQ:**

| 1 | 1 | 1 | 1 | 2 |
|---|---|---|---|---|
| Length = 0x0E | Cmd0 = 0x42 | Cmd1 = 0x82 | Status | DeviceShortAddress |

| 8 | 1 | 1 | 1 |
|---|---|---|---|
| KeySource | SecurityLevel | KeyIdMode | KeyIndex |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | This field indicates either SUCCESS (0) or FAILURE (1). |
| DeviceShortAddress | 2 | Short address of the device |
| KeySource | 8 | Key Source of this data frame. |
| SecurityLevel | 1 | Security Level of this data frame: |

| Security Level | Value |
|---|---|
| NO_SECURITY | 0x00 |
| MIC_32_AUTH | 0x01 |
| MIC_64_AUTH | 0x02 |
| MIC_128_AUTH | 0x03 |
| AES_ENCRYPTION | 0x04 |
| AES_ENCRYPTION_MIC_32 | 0x05 |
| AES_ENCRYPTION_MIC_64 | 0x06 |
| AES_ENCRYPTION_MIC_128 | 0x07 |

Key Id Mode of this data frame:

| Key Id Mode | Value |
|---|---|
| NOT_USED | 0x00 |
| KEY_1BYTE_INDEX | 0x01 |
| KEY_4BYTE_INDEX | 0x02 |
| KEY_8BYTE_INDEX | 0x03 |

KeyIdMode          1

KeyIndex           1          Key Index of this data frame.

### 3.5.2.4    MAC_BEACON_NOTIFY_IND

**Description:**

This callback is called by the MAC to send (on behalf of the next higher layer) a MAC beacon notify indication.

**Usage:**

**AREQ:**

| 1 | 1 | 1 | 1 | 4 | 1 |
|---|---|---|---|---|---|
| Length = 0x24-0xBC | Cmd0 = 0x42 | Cmd1 = 0x83 | BSN | Timestamp | CoordinatorAddressMode |

| 8 | 2 | 2 | 1 | 1 | 1 |
|---|---|---|---|---|---|
| CoordinatorExtendedAddress | PanId | SuperframeSpec | LogicalChannel | GTSPermit | LinkQuality |

| 1 | 8 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|---|
| SecurityFailure | KeySource | SecurityLevel | KeyIdMode | KeyIndex | PendingAddrSpec |

| 1 | 1 | 0-128 |
|---|---|---|
| AddressList | SDULength | NSDU |
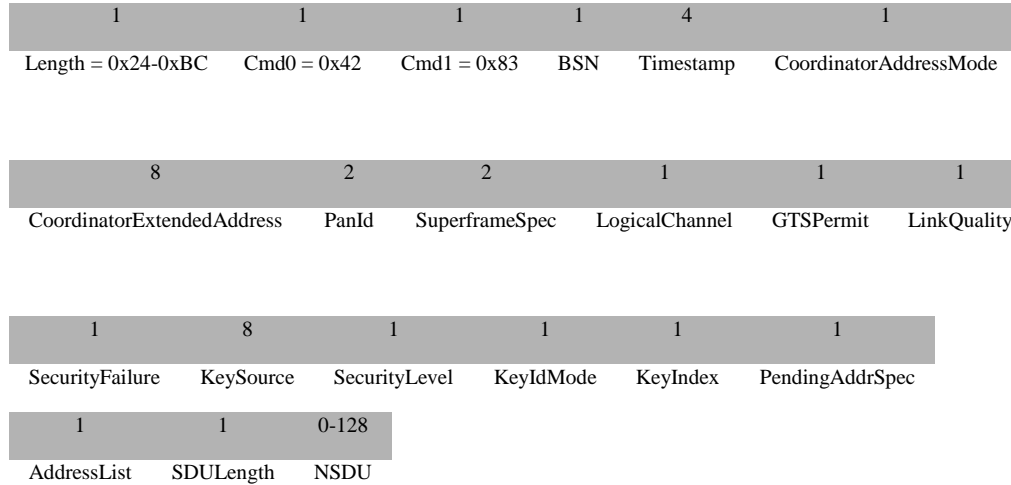
**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| BSN | 1 | BSN |
| Timestamp | 4 | Timestamp of the message |

Address mode of the coordinator

| Mode | Value | Description |
|------|-------|-------------|
| ADDRESS_NOT_PRESENT | 0x00 | Address Not Present |
| GROUP_ADDRESS | 0x01 | Group address |
| ADDRESS_16_BIT | 0x02 | Address 16 bit |
| ADDRESS_64_BIT | 0x03 | Address 64 bit |
| BROADCAST | 0xFF | Broadcast |

CoordinatorAddressMode 1 (aligned with table above)

| | | |
|---|---|---|
| CoordinatorExtendedAddress | 8 | Extended address of the coordinator |
| PanId | 2 | Pan Id of the device |
| SuperframeSpec | 2 | |
| LogicalChannel | 1 | Current logical channel |
| GTSPermit | 1 | TRUE/FALSE - Permit/Not permit GTS |
| LinkQuality | 1 | Link quality of the message |
| SecurityFailure | 1 | |
| KeySource | 8 | Key Source of this data frame. |

Security Level of this data frame:

| Security Level | Value |
|----------------|-------|
| NO_SECURITY | 0x00 |
| MIC_32_AUTH | 0x01 |
| MIC_64_AUTH | 0x02 |
| MIC_128_AUTH | 0x03 |
| AES_ENCRYPTION | 0x04 |
| AES_ENCRYPTION_MIC_32 | 0x05 |
| AES_ENCRYPTION_MIC_64 | 0x06 |
| AES_ENCRYPTION_MIC_128 | 0x07 |

SecurityLevel 1 (aligned with table above)

Key Id Mode of this data frame:

| Key Id Mode | Value |
|-------------|-------|
| NOT_USED | 0x00 |
| KEY_1BYTE_INDEX | 0x01 |
| KEY_4BYTE_INDEX | 0x02 |
| KEY_8BYTE_INDEX | 0x03 |

KeyIdMode 1 (aligned with table above)

| | | |
|---|---|---|
| KeyIndex | 1 | Key Index of this data frame. |
| PendingAddrSpec | 1 | |

| | | |
|---|---|---|
| AddressList | 1 | List of address associate with the device |
| SDULength | 1 | Beacon Length |
| NSDU | 0-128 | Beacon payload |

### 3.5.2.5    MAC_DATA_CNF

**Description:**

This callback is called by the MAC to send (on behalf of the next higher layer) a MAC data confirmation.

**Usage:**

**AREQ:**

| 1 | 1 | 1 | 1 | 1 | 4 | 2 |
|---|---|---|---|---|---|---|
| Length = 0x08 | Cmd0 = 0x42 | Cmd1 = 0x84 | Status | Handle | Timestamp | Timestamp2 |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | This field indicates either SUCCESS (0) or FAILURE (1). |
| Handle | 1 | Handle of the message |
| Timestamp | 4 | 64bit timestamp of the message |
| Timestamp2 | 2 | 16bit timestamp of the message |

### 3.5.2.6    MAC_DATA_IND

**Description:**

This callback is called by the MAC to send (on behalf of the next higher layer) a MAC data indication.

**Usage:**

**AREQ:**

| 1 | 1 | 1 | 1 | 8 | 1 | 8 |
|---|---|---|---|---|---|---|
| Length = 0x2C-0xAC | Cmd0 = 0x42 | Cmd1 = 0x85 | SrcAddrMode | SrcAddr | DstAddrMode | DstAddr |

| 4 | 2 | 2 | 2 | 2 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|
| Timestamp | Timestamp2 | SrcPanId | SrcPanId | DstPanId | LinkQuality | Correlation | RSSI |

| 1 | 8 | 1 | 1 | 1 | 1 | 0-128 |
|---|---|---|---|---|---|---|
| DSN | KeySource | SecurityLevel | KeyIdMode | KeyIndex | Length | Data |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| SrcAddrMode | 1 | Source address mode<br><br>| Mode | Value | Description |<br>|---|---|---|<br>| ADDRESS_NOT_PRESENT | 0x00 | Address Not Present |<br>| GROUP_ADDRESS | 0x01 | Group address |<br>| ADDRESS_16_BIT | 0x02 | Address 16 bit |<br>| ADDRESS_64_BIT | 0x03 | Address 64 bit |<br>| BROADCAST | 0xFF | Broadcast | |
| SrcAddr | 8 | Source address |
| DstAddrMode | 1 | Destination address mode |
| DstAddr | 8 | Destination address |
| Timestamp | 4 | 32bit timestamp of the message |
| Timestamp2 | 2 | 16bit timestamp of the message |
| SrcPanId | 2 | Pan Id of the source address |
| DstPanId | 2 | Pan Id of the destination address |
| LinkQuality | 1 | Link quality |
| Correlation | 1 | Correlation |
| RSSI | 1 | RSSI |
| DSN | 1 | DSN |
| KeySource | 8 | Key Source of this data frame. |
| SecurityLevel | 1 | Security Level of this data frame:<br><br>| Security Level | Value |<br>|---|---|<br>| NO_SECURITY | 0x00 |<br>| MIC_32_AUTH | 0x01 |<br>| MIC_64_AUTH | 0x02 |<br>| MIC_128_AUTH | 0x03 |<br>| AES_ENCRYPTION | 0x04 |<br>| AES_ENCRYPTION_MIC_32 | 0x05 |<br>| AES_ENCRYPTION_MIC_64 | 0x06 | |

| | |
|---|---|
| AES_ENCRYPTION_MIC_128 | 0x07 |

Key Id Mode of this data frame:

| | | Key Id Mode | Value |
|---|---|---|---|
| KeyIdMode | 1 | NOT_USED | 0x00 |
| | | KEY_1BYTE_INDEX | 0x01 |
| | | KEY_4BYTE_INDEX | 0x02 |
| | | KEY_8BYTE_INDEX | 0x03 |
| KeyIndex | 1 | Key Index of this data frame. | |
| Length | 1 | Data length | |
| Data | 0-128 | Data | |

## 3.5.2.7    MAC_DISASSOCIATE_IND

**Description:**

This callback is called by the MAC to send (on behalf of the next higher layer) a MAC disassociation indication.

**Usage:**
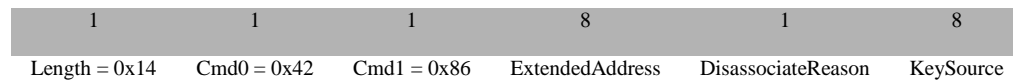
**AREQ:**

| 1 | 1 | 1 | 8 | 1 | 8 |
|---|---|---|---|---|---|
| Length = 0x14 | Cmd0 = 0x42 | Cmd1 = 0x86 | ExtendedAddress | DisassociateReason | KeySource |

| 1 | 1 | 1 |
|---|---|---|
| SecurityLevel | KeyIdMode | KeyIndex |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| ExtendedAddress | 8 | Extended address of the device leaving the network |

Reason of the disassociation:

| DisassociateReason | 1 | Reason | Value |
|---|---|---|---|
| | | Coordinator wishes the device to disassociate | 0x01 |
| | | Device itself wishes to disassociate | 0x02 |

KeySource          8          Key Source of this data frame.

Security Level of this data frame:

| SecurityLevel | 1 | Security Level | Value |
|---|---|---|---|
| | | NO_SECURITY | 0x00 |
| | | MIC_32_AUTH | 0x01 |
| | | MIC_64_AUTH | 0x02 |
| | | MIC_128_AUTH | 0x03 |
| | | AES_ENCRYPTION | 0x04 |
| | | AES_ENCRYPTION_MIC_32 | 0x05 |
| | | AES_ENCRYPTION_MIC_64 | 0x06 |
| | | AES_ENCRYPTION_MIC_128 | 0x07 |

Key Id Mode of this data frame:

| KeyIdMode | 1 | Key Id Mode | Value |
|---|---|---|---|
| | | NOT_USED | 0x00 |
| | | KEY_1BYTE_INDEX | 0x01 |
| | | KEY_4BYTE_INDEX | 0x02 |
| | | KEY_8BYTE_INDEX | 0x03 |

KeyIndex           1          Key Index of this data frame.

### 3.5.2.8    MAC_DISASSOCIATE_CNF

**Description:**

This callback is called by the MAC to send (on behalf of the next higher layer) a MAC disassociate confirm.

**Usage:**

**AREQ:**

| 1 | 1 | 1 | 1 | 1 | 8 | 2 |
|---|---|---|---|---|---|---|
| Length = 0x0C | Cmd0 = 0x42 | Cmd1 = 0x87 | Status | DeviceAddrMode | DeviceAddr | DevicePanId |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | This field indicates either SUCCESS (0) or FAILURE (1). |
| DeviceAddrMode | 1 | Address mode of the device |
| DeviceAddr | 8 | Address of the device |
| DevicePanId | 2 | Pan Id of the device |

| Mode | Value | Description |
|---|---|---|
| ADDRESS_NOT_PRESENT | 0x00 | Address Not Present |
| GROUP_ADDRESS | 0x01 | Group address |
| ADDRESS_16_BIT | 0x02 | Address 16 bit |
| ADDRESS_64_BIT | 0x03 | Address 64 bit |
| BROADCAST | 0xFF | Broadcast |

### 3.5.2.9    MAC_ORPHAN_IND

**Description:**

This callback is called by the MAC to send (on behalf of the next higher layer) a MAC orphan indication.

**Usage:**

**AREQ:**

| 1 | 1 | 1 | 8 |
|---|---|---|---|
| Length = 0x13 | Cmd0 = 0x42 | Cmd1 = 0x8A | ExtendedAddr |

| 8 | 1 | 1 | 1 |
|---|---|---|---|
| KeySource | SecurityLevel | KeyIdMode | KeyIndex |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| ExtendedAddr | 8 | Extended address of the orphan device |
| KeySource | 8 | Key Source of this data frame. |

Security Level of this data frame:

| Security Level | Value |
|---|---|
| NO_SECURITY | 0x00 |
| MIC_32_AUTH | 0x01 |
| MIC_64_AUTH | 0x02 |
| MIC_128_AUTH | 0x03 |
| AES_ENCRYPTION | 0x04 |
| AES_ENCRYPTION_MIC_32 | 0x05 |
| AES_ENCRYPTION_MIC_64 | 0x06 |
| AES_ENCRYPTION_MIC_128 | 0x07 |

SecurityLevel    1

Key Id Mode of this data frame:

| Key Id Mode | Value |
|---|---|
| NOT_USED | 0x00 |
| KEY_1BYTE_INDEX | 0x01 |
| KEY_4BYTE_INDEX | 0x02 |
| KEY_8BYTE_INDEX | 0x03 |

KeyIdMode    1

KeyIndex    1    Key Index of this data frame.

### 3.5.2.10    MAC_POLL_CNF

**Description:**

This callback is called by the MAC to send (on behalf of the next higher layer) a MAC poll confirmation.

**Usage:**
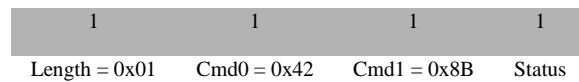
**AREQ:**

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x42 | Cmd1 = 0x8B | Status |

**Attributes**:

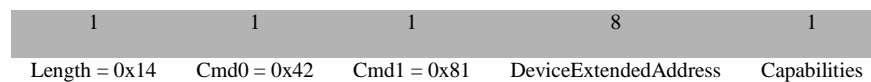| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | This field indicates either SUCCESS (0) or FAILURE (1). |

### 3.5.2.11    MAC_SCAN_CNF

**Description:**

This callback is called by the MAC to send (on behalf of the next higher layer) a MAC scan confirmation.

**Usage:**

**AREQ:**

| 1 | 1 | 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|
| Length = 0x0A-0x8A | Cmd0 = 0x42 | Cmd1 = 0x8C | Status | ED | ScanType | ChannelPage |

| 4 | 1 | 1 | 0-128 |
|---|---|---|---|
| UnscannedChannelList | ResultListCount | ResultListMaxLength | ResultList |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | This field indicates either SUCCESS (0) or FAILURE (1). |
| ED | 1 | ED max energy. |
| ScanType | 1 | Specifies the scan type: <table><tr><td>Scan Type</td><td>Value</td></tr><tr><td>ENERGY_DETECT</td><td>0x00</td></tr><tr><td>ACTIVE</td><td>0x01</td></tr><tr><td>PASSIVE</td><td>0x02</td></tr><tr><td>ORPHAN</td><td>0x03</td></tr></table> |
| ChannelPage | 1 | Channel Page |
| UnscannedChannelList | 4 | List of un-scanned channels |
| ResultListCount | 1 | Number of item in the result list |
| ResultListMaxLength | 1 | Max length of the result list in bytes |
| ResultList | 0-128 | Result list |

### 3.5.2.12    MAC_COMM_STATUS_IND

**Description:**

This callback is called by the MAC to send (on behalf of the next higher layer) a MAC communication indicator.

**Usage:**

**AREQ:**

| 1 | 1 | 1 | 1 | 8 | 1 | 8 |
|---|---|---|---|---|---|---|
| Length = 0x24 | Cmd0 = 0x42 | Cmd1 = 0x8D | Status | SrcAddr | DstAddrMode | DstAddr |

| 4 | 2 | 1 | 8 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|
| Timestamp | DevicePanId | Reason | KeySource | SecurityLevel | KeyIdMode | KeyIndex |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | This field indicates either SUCCESS (0) or FAILURE (1). |
| DstAddrMode | 1 | Destination address mode |
| SrcAddr | 8 | Source address |
| DstAddr | 8 | Destination address |
| Timestamp | 4 | Timestamp of the message |
| DevicePanId | 2 | Pan Id of the device that generate the indication |
| Reason | 1 | Reason for this communication indication. |
| KeySource | 8 | Key Source of this data frame. |
| SecurityLevel | 1 | Security Level of this data frame: |

DstAddrMode table:

| Mode | Value | Description |
|---|---|---|
| ADDRESS_NOT_PRESENT | 0x00 | Address Not Present |
| GROUP_ADDRESS | 0x01 | Group address |
| ADDRESS_16_BIT | 0x02 | Address 16 bit |
| ADDRESS_64_BIT | 0x03 | Address 64 bit |
| BROADCAST | 0xFF | Broadcast |

SecurityLevel table:

| Security Level | Value |
|---|---|
| NO_SECURITY | 0x00 |
| MIC_32_AUTH | 0x01 |
| MIC_64_AUTH | 0x02 |
| MIC_128_AUTH | 0x03 |
| AES_ENCRYPTION | 0x04 |
| AES_ENCRYPTION_MIC_32 | 0x05 |
| AES_ENCRYPTION_MIC_64 | 0x06 |
| AES_ENCRYPTION_MIC_128 | 0x07 |

Key Id Mode of this data frame:

| Key Id Mode | Value |
|---|---|
| NOT_USED | 0x00 |
| KEY_1BYTE_INDEX | 0x01 |
| KEY_4BYTE_INDEX | 0x02 |
| KEY_8BYTE_INDEX | 0x03 |

KeyIdMode              1

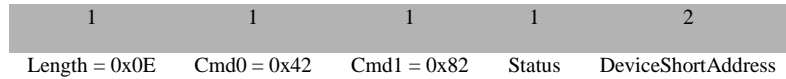KeyIndex              1          Key Index of this data frame.

### 3.5.2.13    MAC_START_CNF

**Description:**

This callback is called by the MAC to send (on behalf of the next higher layer) a MAC start confirmation.

**Usage:**

**AREQ:**

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x42 | Cmd1 = 0x8E | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | This field indicates either SUCCESS (0) or FAILURE (1). |

### 3.5.2.14    MAC_RX_ENABLE_CNF

**Description:**

This callback is called by the MAC to send (on behalf of the next higher layer) a MAC Rx enable confirmation.

**Usage:**

**AREQ:**

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x42 | Cmd1 = 0x8F | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | This field indicates either SUCCESS (0) or FAILURE (1). |

### 3.5.2.15    MAC_PURGE_CNF

**Description:**

This callback is called by the MAC to send (on behalf of the next higher layer) a MAC purge confirmation.

**Usage:**

**AREQ:**

| 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x42 | Cmd1 = 0x9A | Status | Handle |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | This field indicates either SUCCESS (0) or FAILURE (1). |
| Handle | 1 | Handle of this message |

## 3.6    MT_NWK

Not supported.

## 3.7    MT_SAPI

This interface allows tester to interact with the simple API interface.

### 3.7.1    MT_SAPI Commands

#### 3.7.1.1    ZB_SYSTEM_RESET

**Description:**

This command will reset the device by using a soft reset (i.e. a jump to the reset vector) vice a hardware reset (i.e. watchdog reset.) This is especially useful in the CC2531, for instance, so that the USB host does not have to contend with the USB H/W resetting (and thus causing the USB host to re-enumerate the device which can cause an open virtual serial port to hang.)

**Usage:**

**AREQ:**

| Byte: 1 | 1 | 1 |
|---|---|---|
| Length = 0x00 | Cmd0 = 0x46 | Cmd1 = 0x09 |

**Attributes:**

None

### 3.7.1.2     ZB_START_REQUEST

**Description:**

This command starts the ZigBee stack.  When the ZigBee stack starts, the device reads
configuration parameters from nonvolatile memory and the device joins its network.  The ZigBee
stack calls the zb_StartConfirm callback function when the startup process completes. After the
start request process completes, the device is ready to send, receive, and route network traffic.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 |
|---|---|---|
| Length = 0x00 | Cmd0 = 0x26 | Cmd1 = 0x00 |

**Attributes:**

None

**SRSP**:

| Byte: 1 | 1 | 1 |
|---|---|---|
| Length = 0x00 | Cmd0 = 0x66 | Cmd1 = 0x00 |

**Attributes:**

None

### 3.7.1.3     ZB_PERMIT_JOINING_REQUEST

**Description:**

This command is used to control the joining permissions and thus allows or disallows new devices
from joining the network.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 2 | 1 |
|---|---|---|---|---|

| Length = 0x03 | Cmd0 = 0x26 | Cmd1 = 0x08 | Destination | Timeout |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Destination | 2 | The destination parameter indicates the address of the device for which the joining permissions should be set. This is usually the local device address or the special broadcast address that denotes all routers and coordinator (0xFFFC). This way the joining permissions of a single device or the whole network can be controlled. |
| Timeout | 1 | Indicates the amount of time in seconds for which the joining permissions should be turned on. If timeout is set to 0x00, the device will turn off the joining permissions indefinitely. If it is set to 0xFF, the joining permissions will be turned on indefinitely. |

**SRSP:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x66 | Cmd1 = 0x08 | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.7.1.4    ZB_BIND_DEVICE

**Description:**

This command establishes or removes a 'binding' between two devices.  Once bound, an application can send messages to a device by referencing the commandId for the binding.

**Usage:**
**SREQ:**

| 1 | 1 | 1 | 1 | 2 | 8 |
|---|---|---|---|---|---|
| Length = 0x0B | Cmd0 = 0x26 | Cmd1 = 0x01 | Create | CommandId | Destination |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Create | 1 | TRUE to create a binding, FALSE to remove a binding. |
| CommandId | 2 | The Identifier of the binding |
| Destination | 8 | Specifies the 64-bit IEEE address of the device to bind to. |

**SRSP:**

| 1 | 1 | 1 |
|---|---|---|
| Length = 0x00 | Cmd0 = 0x66 | Cmd1 = 0x01 |
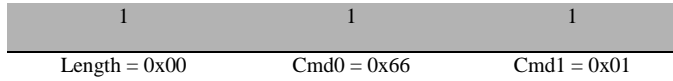
**Attributes:**

None

### 3.7.1.5    ZB_ALLOW_BIND

**Description:**

This command puts the device into the Allow Binding Mode for a given period of time.  A peer device can establish a binding to a device in the Allow Binding Mode by calling zb_BindDevice with a destination address of NULL.

**Usage:**

**SREQ:**

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x26 | Cmd1 = 0x02 | Timeout |
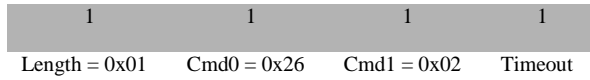
**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Timeout | 1 | The number of seconds to remain in the allow binding mode.  Valid values range from 1 through 65. If 0, the Allow Bind mode will be set false without timeout. If greater than 64, the Allow Bind mode will be true. |

**SRSP:**

| 1 | 1 | 1 |
|---|---|---|
| Length = 0x00 | Cmd0 = 0x66 | Cmd1 = 0x02 |
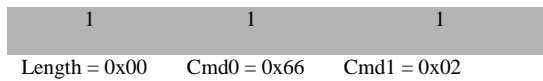
**Attributes:**

None

### 3.7.1.6    ZB_SEND_DATA_REQUEST

**Description:**

This command initiates transmission of data to a peer device.

**Usage:**

**SREQ:**
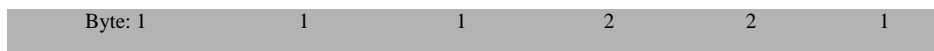
| Byte: 1 | 1 | 1 | 2 | 2 | 1 |
|---|---|---|---|---|---|

| Length = 0x08-0x5C | Cmd0 = 0x26 | Cmd1 = 0x03 | Destination | CommandId | Handle |

| 1 | 1 | 1 | 0-84 |
|---|---|---|---|
| Ack | Radius | Len | Data |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Destination | 2 | The destination of the data.  The destination can be one of the following:<br><br>- 16-Bit short address of device [0-0xfffD]<br><br>- ZB_BROADCAST_ADDR sends the data to all devices in the network.<br><br>- ZB_BINDING_ADDR sends the data to a previously bound device. |
| CommandId | 2 | The command Id to send with the message.  If the ZB_BINDING_ADDR destination is used, this parameter also indicates the binding to use. |
| Handle | 1 | A handle used to Identify the send data request. |
| Ack | 1 | TRUE if requesting acknowledgement from the destination. |
| Radius | 1 | The max number of hops the packet can travel through before it is dropped. |
| Len | 1 | Specifies the size of the Data buffer in bytes. |
| Data | 0-84 | Data |

**SRSP:**

| 1 | 1 | 1 |
|---|---|---|
| Length = 0x00 | Cmd0 = 0x66 | Cmd1 = 0x03 |

**Attributes:**

None

### 3.7.1.7     ZB_READ_CONFIGURATION

**Description:**

This command is used to get a configuration property from non-volatile memory.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x26 | Cmd1 = 0x04 | ConfigId |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| ConfigId | 1 | Specifies the Identifier for the configuration property. |

**SRSP:**

| Byte: 1 | 1 | 1 | 1 | 1 | 1 | 0-128 |
|---|---|---|---|---|---|---|
| Length = 0x03-0x83 | Cmd0 = 0x66 | Cmd1 = 0x04 | Status | ConfigId | Len | Value |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | This field indicates either SUCCESS (0) or FAILURE (1). |
| ConfigId | 1 | Specifies the Identifier for the configuration property. |
| Len | 1 | Specifies the size of the Value buffer in bytes. |
| Value | 0-128 | Buffer to hold the configuration property. |

### 3.7.1.8    ZB_WRITE_CONFIGURATION

**Description:**

This command is used to write a Configuration Property to non-volatile memory.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 1 | 1 | 1-128 |
|---|---|---|---|---|---|
| Length = 0x03-0x83 | Cmd0 = 0x26 | Cmd1 = 0x05 | ConfigId | Len | Value |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| ConfigId | 1 | The Identifier for the configuration property |
| Len | 1 | Specifies the size of the Value buffer in bytes. |
| Value | 1-128 | The buffer containing the new value of the configuration property. |

**SRSP:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x66 | Cmd1 = 0x05 | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | This field indicates either SUCCESS (0) or FAILURE (1). |

## 3.7.1.9    ZB_GET_DEVICE_INFO

**Description:**

This command retrieves a Device Information Property.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x26 | Cmd1 = 0x06 | Param |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Param | 1 | The Identifier for the device information. |

**SRSP:**

| 1 | 1 | 1 | 1 | 8 |
|---|---|---|---|---|
| Length = 0x09 | Cmd0 = 0x66 | Cmd1 = 0x06 | Param | Value |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Param | 1 | The Identifier for the device information. |
| Value | 2 | A buffer to hold the device information |

### 3.7.1.10    ZB_FIND_DEVICE_REQUEST

**Description:**

This command is used to determine the short address for a device in the network.  The device initiating a call to zb_FindDeviceRequest and the device being discovered must both be a member of the same network.  When the search is complete, the zv_FindDeviceConfirm callback function is called.

**SREQ:**

| 1 | 1 | 1 | 8 |
|---|---|---|---|
| Length = 0x08 | Cmd0 = 0x26 | Cmd1 = 0x07 | SearchKey |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| SearchKey | 8 | Specifies the value to search on. |

**SRSP:**

| 1 | 1 | 1 |
|---|---|---|
| Length = 0x00 | Cmd0 = 0x66 | Cmd1 = 0x07 |

**Attributes**:

None

## 3.7.2    MT_SAPI Callbacks

### 3.7.2.1    ZB_START_CONFIRM

**Description:**

This callback is called by the ZigBee stack after a start request operation completes.

**Usage:**

**AREQ:**

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x46 | Cmd1 = 0x80 | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | This field indicates either SUCCESS (0) or FAILURE (1). |

### 3.7.2.2    ZB_BIND_CONFIRM

**Description:**

This callback is called by the ZigBee stack after a bind operation completes.

**Usage:**

**AREQ:**

| 1 | 1 | 1 | 2 | 1 |
|---|---|---|---|---|
| Length = 0x03 | Cmd0 = 0x46 | Cmd1 = 0x81 | CommandId | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| CommandId | 2 | The command Id of the binding being confirmed. |
| Status | 1 | Specifies the status of the bind operation. |

### 3.7.2.3    ZB_ALLOW_BIND_CONFIRM

**Description:**

This callback indicates another device attempted to bind to this device.

**Usage:**

**AREQ:**

| 1 | 1 | 1 | 2 |
|---|---|---|---|
| Length = 0x02 | Cmd0 = 0x46 | Cmd1 = 0x82 | Source |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Source | 2 | Contains the address of the device attempted to bind to this device. |

### 3.7.2.4    ZB_SEND_DATA_CONFIRM

**Description:**

This callback indicates the data has been sent.

**Usage:**

**AREQ:**

| 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|
| Length = 0x02 | Cmd0 = 0x46 | Cmd1 = 0x83 | Handle | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Handle | 1 | Specifies the handle. |
| Status | 1 | This field indicates either SUCCESS (0) or FAILURE (1). |

### 3.7.2.5    ZB_RECEIVE_DATA_INDICATION

**Description:**

This callback is called asynchronously by the ZigBee stack to notify the application when data is received from a peer device.

**Usage:**

**AREQ:**

| 1 | 1 | 1 | 2 | 2 | 2 | 0-84 |
|---|---|---|---|---|---|---|
| Length = 0x06-5A | Cmd0 = 0x46 | Cmd1 = 0x87 | Source | Command | Len | Data |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Source | 2 | Specifies the short address of the peer device that sent the data. |
| Command | 2 | The command Id associated with the data. |
| Len | 2 | Specifies the number of bytes in the Data parameter. |
| Data | 0-84 | The data sent by the peer device. |

### 3.7.2.6    ZB_FIND_DEVICE_CONFIRM

**Description:**

This callback is called by the ZigBee stack when a find device operation completes.

**Usage:**

**AREQ:**

| 1 | 1 | 1 | 1 | 2 | 8 |
|---|---|---|---|---|---|
| Length = 0x0B | Cmd0 = 0x46 | Cmd1 = 0x85 | SearchType = 0x01 | SearchKey | Result |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| SearchType | 1 | The type of search that was performed. |
| SearchKey | 2 | Value that the search was executed on. |
| Result | 8 | The result of the search. |

## 3.8    MT_SYS

This interface allows the tester to interact with the target at system level such as reset, read/write memory, read/write extended address, etc.

### 3.8.1    MT_SYS Commands

### 3.8.1.1    SYS_RESET_REQ

**Description:**

This command is sent by the tester to reset the target device

**Usage:**

**AREQ:**

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x41 | Cmd1 = 0x00 | Type |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Type | 1 | This command will reset the device by using a hardware reset (i.e. watchdog reset) if 'Type' is zero. Otherwise a soft reset (i.e. a jump to the reset vector) is done. This is especially useful in the CC2531, for instance, so that the USB host does not have to contend with the USB H/W resetting (and thus causing the USB host to re-enumerate the device which can cause an open virtual serial port to hang.) |

### 3.8.1.2    SYS_PING

**Description:**

This command issues PING requests to verify if a device is active and check the capability of the device.

**Usage:**

**SREQ:**

| 1 | 1 | 1 |
|---|---|---|
| Length = 0x00 | Cmd0 = 0x21 | Cmd1 = 0x01 |

**Attributes:**

None

**SRSP:**

| 1 | 1 | 1 | 2 |
|---|---|---|---|
| Length = 0x02 | Cmd0 = 0x61 | Cmd1 = 0x01 | Capabilities |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Capabilities | 2 | This field represents the interfaces that this device can handle (compiled into the device). Bit weighted and defined as: |

| Capability | Value |
|---|---|
| MT_CAP_SYS | 0x0001 |
| MT_CAP_MAC | 0x0002 |
| MT_CAP_NWK | 0x0004 |
| MT_CAP_AF | 0x0008 |
| MT_CAP_ZDO | 0x0010 |
| MT_CAP_SAPI | 0x0020 |
| MT_CAP_UTIL | 0x0040 |
| MT_CAP_DEBUG | 0x0080 |
| MT_CAP_APP | 0x0100 |
| MT_CAP_ZOAD | 0x1000 |

### 3.8.1.3    SYS_VERSION

**Description:**

This command is used to request for the device's version string.

**Usage:**

**SREQ:**

| 1 | 1 | 1 |
|---|---|---|
| Length = 0x00 | Cmd0 = 0x21 | Cmd1 = 0x02 |

**Attributes:**

None

**SRSP:**

| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|
| Length = 0x05 | Cmd0 = 0x61 | Cmd1 = 0x02 | TransportRev | Product | MajorRel | MinorRel | MaintRel |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| TransportRev | 1 | Transport protocol revision |
| Product | 1 | Product Id |
| MajorRel | 1 | Software major release number |
| MinorRel | 1 | Software minor release number |
| MaintRel | 1 | Software maintenance release number |

### 3.8.1.4    SYS_SET_EXTADDR

**Description:**

This command is used to set the extended address of the device.

**Usage:**

**SREQ:**

| 1 | 1 | 1 | 8 |
|---|---|---|---|
| Length = 0x08 | Cmd0 = 0x21 | Cmd1 = 0x03 | ExtAddress |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| ExtAddress | 8 | The device's extended address. |

**SRSP:**

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x61 | Cmd1 = 0x03 | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (1) or Failure (0) |

### 3.8.1.5    SYS_GET_EXTADDR

**Description:**

This command is used to get the extended address of the device.

**Usage:**

**SREQ:**

| 1 | 1 | 1 |
|---|---|---|
| Length = 0x00 | Cmd0 = 0x21 | Cmd1 = 0x04 |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (1) or Failure (0) |

**SRSP:**

| 1 | 1 | 1 | 8 |
|---|---|---|---|
| Length = 0x08 | Cmd0 = 0x61 | Cmd1 = 0x04 | ExtAddress |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| ExtAddress | 8 | The device's extended address. |

## 3.8.1.6    SYS_RAM_READ

**Description:**

This command is used by the tester to read a single memory location in the target RAM. The command accepts an address value and returns the memory value present in the target RAM at that address.

**Usage:**
**SREQ:**

| 1 | 1 | 1 | 2 | 1 |
|---|---|---|---|---|
| Length = 0x03 | Cmd0 = 0x21 | Cmd1 = 0x05 | Address | Len |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Address | 2 | Address of the memory that will be read. |
| Len | 1 | The number of bytes that will be read from the target RAM. |

**SRSP:**

| 1 | 1 | 1 | 1 | 1 | 0-128 |
|---|---|---|---|---|-------|
| Length = 0x02-0x82 | Cmd0 = 0x61 | Cmd1 = 0x05 | Status | Len | Value |

**Attributes:**

| Attribute | Length (byte) | Description |
|-----------|---------------|-------------|
| Status | 1 | Status is either Success (0) or Failure (1). |
| Len | 1 | The number of bytes that will be read from the target RAM. |
| Value | 0-128 | The value read from the target RAM. |

## 3.8.1.7    SYS_RAM_WRITE

**Description:**

This command is used by the tester to write to a particular location in the target RAM. The command accepts an address location and a memory value. The memory value is written to the address location in the target RAM.

**Usage:**

**SREQ**:

| 1 | 1 | 1 | 2 | 1 | 1-128 |
|---|---|---|---|---|-------|
| Length = 0x04-0x84 | Cmd0 = 0x21 | Cmd1 = 0x06 | Address | Len | Value |

**Attributes:**

| Attribute | Length (byte) | Description |
|-----------|---------------|-------------|
| Address | 2 | Address of the memory that will be read. |
| Len | 1 | The number of bytes that will be read from the target RAM. |
| Value | 1-128 | The value written to the target RAMS. |

**SRSP**:

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x61 | Cmd1 = 0x06 | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.8.1.8    SYS_OSAL_NV_READ

**Description:**

This command is used by the tester to read a single memory item from the target non-volatile memory. The command accepts an attribute Id value and data offset and returns the memory value present in the target for the specified attribute Id.

**Usage:**

**SREQ:**

| 1 | 1 | 1 | 2 | 1 |
|---|---|---|---|---|
| Length = 0x03 | Cmd0 = 0x21 | Cmd1 = 0x08 | Id | Offset |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Id | 2 | The Id of the NV item. |
| Offset | 1 | Number of bytes offset from the beginning or the NV value. |

**SRSP:**

| 1 | 1 | 1 | 1 | 1 | 0-248 |
|---|---|---|---|---|---|
| Length = 0x02-0xFA | Cmd0 = 0x61 | Cmd1 = 0x08 | Status | Len | Value |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |
| Len | 1 | Length of the NV value. |
| Value | 0-248 | Value of the NV item. |

### 3.8.1.9    SYS_OSAL_NV_WRITE

**Description:**

This command is used by the tester to write to a particular item in non-volatile memory. The command accepts an attribute Id, data offset, data length, and attribute value. The attribute value is written to the location specified for the attribute Id in the target.

**Usage:**

**SREQ:**

| 1 | 1 | 1 | 2 | 1 | 1 | 1-246 |
|---|---|---|---|---|---|---|
| Length = 0x04-0xFA | Cmd0 = 0x21 | Cmd1 = 0x09 | Id | Offset | Len | Value |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Id | 2 | The Id of the NV item. |
| Offset | 1 | Number of bytes offset from the beginning or the NV value. |
| Len | 1 | Length of the NV value. |
| Value | 0-246 | Value of the NV item. |

**SRSP:**

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x61 | Cmd1 = 0x09 | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.8.1.10    SYS_OSAL_NV_ITEM_INIT

**Description**

This command is used by the tester to create and initialize an item in non-volatile memory. The NV item will be created if it does not already exist. The data for the new NV item will be left uninitialized if the *InitLen* parameter is zero. When *InitLen* is non-zero, the data for the NV item will be initialized (starting at offset of zero) with the values from *InitData*. Note that it is not necessary to initialize the entire NV item (*InitLen* < *ItemLen*). It is also possible to create an NV item that is larger than the maximum length *InitData* – use the SYS_OSAL_NV_WRITE command to finish the initialization.

**Usage**

**SREQ:**

| 1 | 1 | 1 | 2 | 2 | 1 | 0-245 |
|---|---|---|---|---|---|---|
| Length = 0x04-0xFA | Cmd0 = 0x21 | Cmd1 = 0x07 | Id | ItemLen | InitLen | InitData |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Id | 2 | The Id of the NV item. |
| ItemLen | 2 | Number of bytes in the NV item. |
| InitLen | 1 | Number of bytes in the initialization data. |
| InitData | 0-245 | Value of the initialization data. |

**SRSP:**

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x61 | Cmd1 = 0x07 | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | 0x00 = Item already exists, no action taken |
| | | 0x09 = Success, item created and initialized |
| | | 0x0A = Initialization failed, item not created |

### 3.8.1.11    SYS_OSAL_NV_DELETE

**Description**

This command is used by the tester to delete an item from the non-volatile memory. The *ItemLen* parameter must match the length of the NV item or the command will fail. Use this command with caution – deleted items cannot be recovered.

**Usage**

**SREQ:**

| 1 | 1 | 1 | 2 | 2 |
|---|---|---|---|---|
| Length = 0x04 | Cmd0 = 0x21 | Cmd1 = 0x12 | Id | ItemLen |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Id | 2 | The Id of the NV item. |
| ItemLen | 2 | Length of the NV item. |

**SRSP:**

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x61 | Cmd1 = 0x12 | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | 0x00 = Success, the item was deleted<br>0x09 = No action taken, item did not exist<br>0x0A = Failure, item was not deleted<br>0x0C = Bad length, item was not deleted |

### 3.8.1.12    SYS_OSAL_NV_LENGTH

**Description**

This command is used by the tester to get the length of an item in non-volatile memory. A returned length of zero indicates that the NV item does not exist.

**Usage**

**SREQ:**

| 1 | 1 | 1 | 2 |
|---|---|---|---|
| Length = 0x02 | Cmd0 = 0x21 | Cmd1 = 0x13 | Id |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Id | 2 | The Id of the NV item. |

**SRSP:**

| 1 | 1 | 1 | 2 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x61 | Cmd1 = 0x13 | Length |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Length | 2 | 0x0000 = item does not exist |
| | | 0x0001-0xNNNN = number of bytes in NV item |

### 3.8.1.13    SYS_OSAL_START_TIMER

**Description**

This command is used by the tester to start a timer event. The event will expired after the indicated amount of time and a notification will be sent back to the tester.

**Usage**

**SREQ:**

| 1 | 1 | 1 | 1 | 2 |
|---|---|---|---|---|
| Length = 0x03 | Cmd0 = 0x21 | Cmd1 = 0x0A | Id | Timeout |

**Attributes:**

| Attribute | Length (byte) | Description |
| --- | --- | --- |
| Id | 1 | The Id of the timer event (0-3) |
| Timeout | 2 | Amount of time it will take before the event expired in milliseconds. |

**SRSP:**

| 1 | 1 | 1 | 1 |
| --- | --- | --- | --- |
| Length = 0x01 | Cmd0 = 0x61 | Cmd1 = 0x0A | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
| --- | --- | --- |
| Status | 1 | Status is either Success (0) or Failure (1). |

## 3.8.1.14    SYS_OSAL_STOP_TIMER

**Description:**

This command is used by the tester to stop a timer event.

**Usage:**

**SREQ:**

| 1 | 1 | 1 | 1 |
| --- | --- | --- | --- |
| Length = 0x01 | Cmd0 = 0x21 | Cmd1 = 0x0B | Id |

**Attributes:**

| Attribute | Length (byte) | Description |
| --- | --- | --- |
| Id | 1 | The Id of the timer event (0-3). |

**SRSP:**

| 1 | 1 | 1 | 1 |
| --- | --- | --- | --- |
| Length = 0x01 | Cmd0 = 0x61 | Cmd1 = 0x0B | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
| --- | --- | --- |

Z-Stack Monitor and Test API                                    SWRA198  Revision 1.17

| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.8.1.15   SYS_RANDOM

**Description:**

This command is used by the tester to get a random 16-bit number.

**Usage:**

**SREQ:**

| 1 | 1 | 1 |
|---|---|---|
| Length = 0x00 | Cmd0 = 0x21 | Cmd1 = 0x0C |

**Attributes:**

None

**SRSP:**

| 1 | 1 | 1 | 2 |
|---|---|---|---|
| Length = 0x02 | Cmd0 = 0x61 | Cmd1 = 0x0C | Value |

**Attributes:**

| Attribute | Length (byte) | Description |
|-----------|---------------|-------------|
| Value | 2 | The random value. |

### 3.8.1.16   SYS_ADC_READ

**Description**

This command reads a value from the ADC based on specified channel and resolution.

**Usage**

**SREQ:**

| 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|
| Length = 0x02 | Cmd0 = 0x21 | Cmd1 = 0x0D | Channel | Resolution |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| | | The channel of the ADC that will be used. |

| Channel | Value |
|---|---|
| AIN0 | 0x00 |
| AIN1 | 0x01 |
| AIN2 | 0x02 |
| AIN3 | 0x03 |
| AIN4 | 0x04 |
| AIN5 | 0x05 |
| AIN6 | 0x06 |
| AIN7 | 0x07 |
| Temperature Sensor | 0x0E |
| Voltage Reading | 0x0F |

Channel — 1

Resolution of the reading. This can be 8-bit, 10-bit, 12-bit or 14-bit.

| Resolution | Value |
|---|---|
| 8-bit | 0x00 |
| 10-bit | 0x01 |
| 12-bit | 0x02 |
| 14-bit | 0x03 |

Resolution — 1

**SRSP**:

| 1 | 1 | 1 | 2 |
|---|---|---|---|
| Length = 0x02 | Cmd0 = 0x61 | Cmd1 = 0x0D | Value |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Value | 2 | Value of the ADC reading based on the specified information. |

### 3.8.1.17    SYS_GPIO

**Description**

This command is used by the tester to control the 4 GPIO pins on the CC2530-ZNP build.

**Usage**

**SREQ:**

| 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|
| Length = 0x02 | Cmd0 = 0x21 | Cmd1 = 0x0E | Operation | Value |

Operation – 1 byte – specifies the type of operation to perform on the GPIO pins. It can take the values, shown in the table below, with effects dictated by the bit values of the value parameter:

**Attributes:**

| Operation | Description |
|---|---|
| Set direction ( 0x00 ) | Configure the direction of the GPIO pins. A value of 0 in a bit position configures the corresponding GPIO pin as an Input while a value of 1 configures it as Output. |
| Set Input mode ( 0x01 ) | Configure the Input mode of the GPIO pins. A value of 0 in a bit position configures it as pull-up mode while a 1 configures it in tri-state Input mode. ( *Note: P1_0 and P1_1 of the CC2530 can only be set in tri-state input mode* ). |
| Set ( 0x02 ) | A value of 1 in a bit position will set the corresponding GPIO pin ( writes a 1 ). |
| Clear ( 0x03 ) | A value of 0 in a bit position will clear the corresponding GPIO pin ( writes a 0 ). |
| Toggle ( 0x04 ) | A value of 1 in a bit position will toggle the corresponding GPIO pin. |
| Read ( 0x05 ) | Reads the GPIO pins. |

**SRSP**:

| 1 | 1 | 1 | 2 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x61 | Cmd1 = 0x0E | Value |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Value | 1 | The value read from the GPIO pins. |

### 3.8.1.18   SYS_STACK_TUNE

**Description**

This command is used by the tester to tune stack parameters and adjust performance at runtime.

**Usage**

**SREQ:**

| 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|
| Length = 0x02 | Cmd0 = 0x21 | Cmd1 = 0x0F | Operation | Value |

**Attributes:**

The tuning operation to be executed according to the STK_Tune_t enumeration:

| Operation | Value |
|---|---|
| Set the transmitter power level according to the value of the Value parameter which should correspond to the valid values specified by the ZMacTransmitPower_t enumeration  (0xFD – 0x16) | 0x00 |
| Set RxOnWhenIdle off/on if the value of Value is 0/1; otherwise return the current setting of RxOnWhenIdle. | 0x01 |

**SRSP**:

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x61 | Cmd1 = 0x0F | Value |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Value | 1 | Applicable status of the tuning operation. |

### 3.8.1.19   SYS_SET_TIME

**Description**

This command is used by the tester to set the target system date and time. The time can be specified in "seconds since 00:00:00 on January 1, 2000" or in parsed date/time components.

**Usage**

**SREQ:**

| 1 | 1 | 1 | 4 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|
| Length = 0x0B | Cmd0 = 0x21 | Cmd1 = 0x10 | UTCTime | Hour | Minute | Second |

| 1 | 1 | 2 |
|---|---|---|
| Month | Day | Year |

**Attributes:**

| Attribute | Length (bytes) | Description |
|---|---|---|
| UTCTime | 4 | Number of seconds since 00:00:00 on January 1, 2000.<br><br>Set this parameter to zero to use date/time attributes below. |
| Hour | 1 | Hour of the day (0-23) |
| Minute | 1 | Minute of the hour (0-59) |
| Second | 1 | Second of the minute (0-59) |
| Month | 1 | Month of the year (1-12) |
| Day | 1 | Day of the month (1-31) |
| Year | 2 | Year (2000- ) |

**SRSP**:

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x61 | Cmd1 = 0x10 | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.8.1.20    SYS_GET_TIME

**Description:**

This command is used by the tester to get the target system date and time. The time is returned in "seconds since 00:00:00 on January 1, 2000" and parsed date/time components.

**Usage:**

**SREQ:**

| 1 | 1 | 1 |
|---|---|---|
| Length = 0x00 | Cmd0 = 0x21 | Cmd1 = 0x11 |

**Attributes:**

None

**SRSP:**

| 1 | 1 | 1 | 4 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|
| Length = 0x0B | Cmd0 = 0x61 | Cmd1 = 0x11 | UTCTime | Hour | Minute | Second |

| 1 | 1 | 2 |
|---|---|---|
| Month | Day | Year |

**Attributes:**

| Attribute | Length (bytes) | Description |
|---|---|---|
| UTCTime | 4 | Number of seconds since 00:00:00 on January 1, 2000. |
| Hour | 1 | Hour of the day (0-23) |
| Minute | 1 | Minute of the hour (0-59) |
| Second | 1 | Second of the minute (0-59) |
| Month | 1 | Month of the year (1-12) |
| Day | 1 | Day of the month (1-31) |
| Year | 2 | Year (2000- ) |

### 3.8.1.21    SYS_SET_TX_POWER

**Description:**

This command is used by the tester to set the target system radio transmit power. The returned TX power is the actual setting applied to the radio – nearest characterized value for the specific radio.

**Usage:**

**SREQ:**

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x21 | Cmd1 = 0x14 | TX Power |

**Attributes:**

| Attribute | Length (bytes) | Description |
|---|---|---|
| TX Power | 1 | Requested TX power setting, in dBm. |

**SRSP**:

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x61 | Cmd1 = 0x14 | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

## 3.8.1.22    SYS_ZDIAGS_INIT_STATS

**Description:**

This command is used to initialize the statistics table in NV memory.

**Usage:**

**SREQ:**

| 1 | 1 | 1 |
|---|---|---|
| Length = 0x00 | Cmd0 = 0x21 | Cmd1 = 0x17 |

**Attributes:**

None

**SRSP:**

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x61 | Cmd1 = 0x17 | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

## 3.8.1.23    SYS_ZDIAGS_CLEAR_STATS

**Description:**

This command is used to clear the statistics table. To clear data in NV (including the Boot Counter) the clearNV flag shall be set to TRUE.

**Usage:**

**SREQ:**

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x21 | Cmd1 = 0x18 | clearNV |

**Attributes:**

| Attribute | Length (bytes) | Description |
|---|---|---|
| clearNV | 1 | TRUE – Clears statistics in NV memory including Boot Counter.<br><br>FALSE – Clears statistics in RAM only. Boot Counter is preserved. |

**SRSP:**

| 1 | 1 | 1 | 4 |
|---|---|---|---|
| Length = 0x04 | Cmd0 = 0x61 | Cmd1 = 0x18 | SysClock |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| SysClock | 4 | Milliseconds since last reset. |

### 3.8.1.24    SYS_ZDIAGS_GET_STATS

**Description:**

This command is used to read a specific system (attribute) ID statistics and/or metrics value.

**Usage:**

**SREQ:**

| 1 | 1 | 1 | 2 |
|---|---|---|---|
| Length = 0x02 | Cmd0 = 0x21 | Cmd1 = 0x19 | AttributeID |

**Attributes:**

| Attribute | Length (bytes) | Description |
|---|---|---|
| AttributeID | 2 | System Diagnostics (ZDiags) attribute ID, as defined in ZDiags.h module. |

**SRSP:**

| 1 | 1 | 1 | 4 |
|---|---|---|---|
| Length = 0x04 | Cmd0 = 0x61 | Cmd1 = 0x19 | AttributeValue |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| AttributeValue | 4 | Value of the requested attribute. |

### 3.8.1.25    SYS_ZDIAGS_RESTORE_STATS_NV

**Description:**

This command is used to restore the statistics table from NV into the RAM table.

**Usage:**

**SREQ:**

| 1 | 1 | 1 |
|---|---|---|
| Length = 0x00 | Cmd0 = 0x21 | Cmd1 = 0x1A |

**Attributes:**

None

**SRSP:**

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x61 | Cmd1 = 0x1A | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

## 3.8.1.26    SYS_ZDIAGS_SAVE_STATS_TO_NV

**Description:**

This command is used to save the statistics table from RAM to NV.

**Usage:**

**SREQ:**

| 1 | 1 | 1 |
|---|---|---|
| Length = 0x00 | Cmd0 = 0x21 | Cmd1 = 0x1B |

**Attributes:**

None

**SRSP:**

| 1 | 1 | 1 | 4 |
|---|---|---|---|
| Length = 0x04 | Cmd0 = 0x61 | Cmd1 = 0x1B | SysClock |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| SysClock | 4 | Milliseconds since last reset. |

### 3.8.1.27   SYS_NV_CREATE

**Description:**

This command is used to attempt to create an item in non-volatile memory.

**Usage:**

**SREQ:**

| 1 | 1 | 1 | 1 | 2 | 2 | 4 |
|---|---|---|---|---|---|---|
| Length = 0x09 | Cmd0 = 0x21 | Cmd1 = 0x30 | SysID | ItemID | SubID | Length |

**Attributes:**

| Attribute | Length (bytes) | Description |
|---|---|---|
| SysID | 1 | System ID of the NV item |
| ItemID | 2 | Item ID of the NV item |
| SubID | 2 | Sub ID of the NV item |
| Length | 4 | Number of bytes in the NV item |

**SRSP:**

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x61 | Cmd1 = 0x30 | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status of NV create |

### 3.8.1.28   SYS_NV_DELETE

**Description:**

This command is used to attempt to delete an item in non-volatile memory.

**Usage:**

**SREQ:**

| 1 | 1 | 1 | 1 | 2 | 2 |
|---|---|---|---|---|---|
| Length = 0x05 | Cmd0 = 0x21 | Cmd1 = 0x31 | SysID | ItemID | SubID |

**Attributes:**

| Attribute | Length (bytes) | Description |
|-----------|----------------|-------------|
| SysID | 1 | System ID of the NV item |
| ItemID | 2 | Item ID of the NV item |
| SubID | 2 | Sub ID of the NV item |

**SRSP:**

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x61 | Cmd1 = 0x31 | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|-----------|---------------|-------------|
| Status | 1 | Status of NV delete |

### 3.8.1.29    SYS_NV_LENGTH

**Description:**

This command is used to get the length of an item in non-volatile memory.

**Usage:**

**SREQ:**

| 1 | 1 | 1 | 1 | 2 | 2 |
|---|---|---|---|---|---|
| Length = 0x05 | Cmd0 = 0x21 | Cmd1 = 0x32 | SysID | ItemID | SubID |

**Attributes:**

| Attribute | Length (bytes) | Description |
|-----------|----------------|-------------|
| SysID | 1 | System ID of the NV item |
| ItemID | 2 | Item ID of the NV item |
| SubID | 2 | Sub ID of the NV item |

**SRSP:**

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x61 | Cmd1 = 0x32 | Length |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Length | 1 | Length of NV Item |

### 3.8.1.30    SYS_NV_READ

**Description:**

This command is used to read an item in non-volatile memory.

**Usage:**

**SREQ:**

| 1 | 1 | 1 | 1 | 2 | 2 | 2 | 1 |
|---|---|---|---|---|---|---|---|
| Length = 0x08 | Cmd0 = 0x21 | Cmd1 = 0x33 | SysID | ItemID | SubID | Offset | Length |

**Attributes:**

| Attribute | Length (bytes) | Description |
|---|---|---|
| SysID | 1 | System ID of the NV item |
| ItemID | 2 | Item ID of the NV item |
| SubID | 2 | Sub ID of the NV item |
| Offset | 2 | Offset into NV |
| Length | 1 | Length of data to read |

**SRSP:**

| 1 | 1 | 1 | 1 | 1 | 0-240 |
|---|---|---|---|---|---|
| Length = 0x02-0xF2 | Cmd0 = 0x61 | Cmd1 = 0x33 | Status | Length | Value |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |
| Length | 1 | Length of NV Item |
| Value | 0-240 | Value of the NV Item read |

### 3.8.1.31   SYS_NV_WRITE

**Description:**

This command is used to write an item in non-volatile memory

**Usage:**

**SREQ:**

| 1 | 1 | 1 | 1 | 2 | 2 | 2 | 1 | 0-240 |
|---|---|---|---|---|---|---|---|---|
| Length = 0x08-0xF8 | Cmd0 = 0x21 | Cmd1 = 0x34 | SysID | ItemID | SubID | Offset | Length | Value |

**Attributes:**

| Attribute | Length (bytes) | Description |
|---|---|---|
| SysID | 1 | System ID of the NV item |
| ItemID | 2 | Item ID of the NV item |
| SubID | 2 | Sub ID of the NV item |
| Offset | 2 | Offset into NV |
| Length | 1 | Length of data to read |
| Value | 0-240 | Data Bytes to be written to NV item |

**SRSP:**

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x61 | Cmd1 = 0x34 | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.8.1.32   SYS_NV_UPDATE

**Description:**

This command is used to update an item in non-volatile memory

**Usage:**

**SREQ:**

| 1 | 1 | 1 | 1 | 2 | 2 | 1 | 0-240 |
|---|---|---|---|---|---|---|---|
| Length = 0x06-0xF6 | Cmd0 = 0x21 | Cmd1 = 0x35 | SysID | ItemID | SubID | Length | Value |

**Attributes:**

| Attribute | Length (bytes) | Description |
|---|---|---|
| SysID | 1 | System ID of the NV item |
| ItemID | 2 | Item ID of the NV item |
| SubID | 2 | Sub ID of the NV item |
| Length | 1 | Length of data to read |
| Value | 0-240 | Data Bytes to be written to NV item |

**SRSP:**

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x61 | Cmd1 = 0x35 | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.8.1.33   SYS_NV_COMPACT

**Description:**

This command is used to compact the active page in non-volatile memory

**Usage:**

**SREQ:**

| 1 | 1 | 1 | 2 |
|---|---|---|---|
| Length = 0x02 | Cmd0 = 0x21 | Cmd1 = 0x36 | Threshold |

**Attributes:**

| Attribute | Length (bytes) | Description |
|-----------|----------------|-------------|
| Threshold | 2 | Compaction occurs when NV bytes are less than this value |

**SRSP:**

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x61 | Cmd1 = 0x36 | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|-----------|---------------|-------------|
| Status | 1 | Status of compaction of active page |

### 3.8.1.34    SYS_OSAL_NV_READ_EXT

**Description:**

This command is used by the tester to read a single memory item from the target non-volatile memory. The command accepts an attribute Id value and data offset and returns the memory value present in the target for the specified attribute Id.

**Usage:**

**SREQ:**

| 1 | 1 | 1 | 2 | 2 |
|---|---|---|---|---|
| Length = 0x03 | Cmd0 = 0x21 | Cmd1 = 0x08 | Id | Offset |

**Attributes:**

| Attribute | Length (byte) | Description |
|-----------|---------------|-------------|
| Id | 2 | The Id of the NV item. |
| Offset | 2 | Number of bytes offset from the beginning or the NV value. |

**SRSP:**

| 1 | 1 | 1 | 1 | 1 | 0-248 |
|---|---|---|---|---|-------|
| Length = 0x02-0xFA | Cmd0 = 0x61 | Cmd1 = 0x08 | Status | Len | Value |

**Attributes:**

| Attribute | Length (byte) | Description |
|-----------|---------------|-------------|
| Status | 1 | Status is either Success (0) or Failure (1). |
| Len | 1 | Length of the NV value. |
| Value | 0-248 | Value of the NV item. |

### 3.8.1.35   SYS_OSAL_NV_WRITE_EXT

**Description:**

This command is used by the tester to write to a particular item in non-volatile memory. The command accepts an attribute Id, data offset, data length, and attribute value. The attribute value is written to the location specified for the attribute Id in the target.

**Usage:**

**SREQ:**

| 1 | 1 | 1 | 2 | 2 | 1 | 1-246 |
|---|---|---|---|---|---|-------|
| Length = 0x04-0xFA | Cmd0 = 0x21 | Cmd1 = 0x09 | Id | Offset | Len | Value |

**Attributes:**

| Attribute | Length (byte) | Description |
|-----------|---------------|-------------|
| Id | 2 | The Id of the NV item. |
| Offset | 2 | Number of bytes offset from the beginning or the NV value. |
| Len | 1 | Length of the NV value. |
| Value | 0-246 | Value of the NV item. |

**SRSP:**

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x61 | Cmd1 = 0x09 | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|-----------|---------------|-------------|
| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.8.2    MT_SYS Callbacks

### 3.8.2.1    SYS_RESET_IND

**Description**

This callback is sent by the device to indicate that a reset has occurred.

**Usage**

**AREQ:**

| 1 | 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|---|
| Length = 0x06 | Cmd0 = 0x41 | Cmd1 = 0x80 | Reason | TransportRev | ProductId |

| 1 | 1 | 1 |
|---|---|---|
| MajorRel | MinorRel | HwRev |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Reason | 1 | Reason for the reset. <table><tr><th>Resolution</th><th>Value</th></tr><tr><td>Power-up</td><td>0x00</td></tr><tr><td>External</td><td>0x01</td></tr><tr><td>Watch-dog</td><td>0x02</td></tr></table> |
| TransportRev | 1 | Transport protocol revision. |
| Product | 1 | Major release number. |
| MinorRel | 1 | Minor release number. |
| HwRev | 1 | Hardware revision number. |

### 3.8.2.2    SYS_OSAL_TIMER_EXPIRED

**Description:**

This callback is sent by the device to indicate that a specific timer has been expired.

**Usage:**

**AREQ:**

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x41 | Cmd1 = 0x81 | Id |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Id | 1 | The Id of the timer event (0-3) |

## 3.9    MT_UART

This interface handles communication between the target and Z-Tool. Tester has no direct control over this interface. There is no direct command for the tester to interact with this interface.

## 3.10    MT_UTIL

This interface provides tester supporting functionalities such as setting PanId, getting device info, getting NV info, subscribing callbacks…etc.

### 3.10.1   MT_UTIL Commands

#### 3.10.1.1    UTIL_GET_DEVICE_INFO

**Description:**

This command is sent by the tester to retrieve the device info.

**Usage:**

**SREQ:**

| 1 | 1 | 1 |
|---|---|---|
| Length = 0x00 | Cmd0 = 0x27 | Cmd1 = 0x00 |

**Attributes:**

None

**SRSP**:

| 1 | 1 | 1 | 1 | 8 | 2 | 1 |
|---|---|---|---|---|---|---|
| Length = 0x02 | Cmd0 = 0x67 | Cmd1 = 0x00 | Status | IEEEAddr | ShortAddr | DeviceType |

| 1 | 1 | 0-128 |
|---|---|---|
| DeviceState | NumAssocDevices | AssocDeviceList |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| | | |

| Status | 1 | Status is a one byte field and is either success (0) or fail (1). The fail status is returned if the address value in the command message was not within the valid range. |
|---|---|---|
| IEEEAddr | 8 | IEEE address of the device |
| ShortAddr | 2 | Short address of the device |
| DeviceType | 1 | Indicates device type, where bits 0 to 2 indicate the capability for the device to operate as a coordinator, router, or end device, respectively. |
| DeviceState | 1 | Indicates the state of the device with different possible states as shown below: 0x00: Initialized - not started automatically 0x01: Initialized - not connected to anything 0x02: Discovering PAN's to join 0x03: Joining a PAN 0x04: Rejoining a PAN, only for end devices 0x05: Joined but not yet authenticated by trust center 0x06: Started as device after authentication 0x07: Device joined, authenticated and is a router 0x08: Starting as ZigBee Coordinator 0x09: Started as ZigBee Coordinator 0x0A: Device has lost information about its parent |
| NumAssocDevices | 1 | Specifies the number of Reduce Function Devices being associated to the target device. |
| AssocDevicesList | Array | Array of 16-bits of network addresses of Reduce Function Devices associated to the local device. |

## 3.10.1.2   UTIL_GET_NV_INFO

**Description:**

This command is used by the tester to read a block of parameters from non-volatile storage of the target device.

**Usage:**

**SREQ:**

| 1 | 1 | 1 |
|---|---|---|
| Length = 0x00 | Cmd0 = 0x27 | Cmd1 = 0x01 |

**Attributes:**

None

**SRSP**:

| 1 | 1 | 1 | 1 | 8 | 4 | 2 |
|---|---|---|---|---|---|---|
| Length = 0x20 | Cmd0 = 0x67 | Cmd1 = 0x01 | Status | IEEEAddr | ScanChannels | PanId |

| 1 | 16 |
|---|---|
| SecurityLevel | PreConfigKey |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | A value of zero indicates success. Failure is indicated by a non-zero value, representing a bit mask of each item that failed to be retrieved from NV memory. Bit0 is used for the first item (IEEEAddress), bit1 for the second item (ScanChannels), and so forth. Data values for failed items are returned as one or more bytes of 0xFF, the typical value read from erased NV memory. |
| IEEEAddr | 8 | IEEE address of the device |
| ScanChannels | 4 | This represents a bit-mask of channels to be scanned when starting the device. |
| PanId | 2 | Specifies the Pan Id to start or join. Set to 0xFFFF to select a PAN after scanning. |
| SecurityLevel | 1 | This specifies the network messaging security level, zero disables security. |
| PreConfigKey | 16 | This specifies the pre-configured security key. |

### 3.10.1.3    UTIL_SET_PANID

**Description:**

Store a PanId value into non-volatile memory to be used the next time the target device resets.

**Usage:**

**SREQ:**

| 1 | 1 | 1 | 2 |
|---|---|---|---|
| Length = 0x02 | Cmd0 = 0x27 | Cmd1 = 0x02 | PanId |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| PanId | 2 | PanId that will be set |

**SRSP:**

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x67 | Cmd1 = 0x02 | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.10.1.4    UTIL_SET_CHANNELS

**Description:**

This command is used to store a channel select bit-mask into non-volatile memory to be used the next time the target device resets.

**Usage:**

**SREQ:**

| 1 | 1 | 1 | 4 |
|---|---|---|---|
| Length = 0x04 | Cmd0 = 0x27 | Cmd1 = 0x03 | Channels |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Channels | 4 | A bit-mask representing the channel(s) to scan the next time the target device resets. |

**SRSP:**

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x67 | Cmd1 = 0x03 | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.10.1.5   UTIL_SET_SECLEVEL

**Description:**

This command is used to store a security level value into non-volatile memory to be used the next time the target device resets.

**Usage:**

**SREQ:**

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x27 | Cmd1 = 0x04 | SecLevel |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| SecLevel | 1 | Security level to use the next time the target device resets. Zero is used to disable security. |

**SRSP:**

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x67 | Cmd1 = 0x04 | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.10.1.6   UTIL_SET_PRECFGKEY

**Description:**

This command is used to store a pre-configured key array into non-volatile memory to be used the next time the target device resets.

**Usage:**

**SREQ:**

| 1 | 1 | 1 | 16 |
|---|---|---|---|
| Length = 0x10 | Cmd0 = 0x27 | Cmd1 = 0x05 | PreCfgKey |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| PreCfgKey | 16 | An array representing the pre-configured key to use the next time the target device resets. |

**SRSP:**

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x67 | Cmd1 = 0x05 | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.10.1.7   UTIL_CALLBACK_SUB_CMD

**Description:**

This command subscribes/unsubscribes to layer callbacks. For particular subsystem callbacks to work, the software must be compiled with a special flag that is unique to that subsystem to enable the callback mechanism. For example to enable ZDO callbacks, MT_ZDO_CB_FUNC flag must be compiled when the software is built. For complete list of callback compile flags, check section 1.2 or "**Z-Stack Compile Options**" document.

**Usage:**

**SREQ:**

| 1 | 1 | 1 | 2 | 1 |
|---|---|---|---|---|
| Length = 0x03 | Cmd0 = 0x27 | Cmd1 = 0x06 | SubsystemId | Action |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| | | Subsystem Id of the expected layer |

| Subsystem | Id |
|---|---|
| MT_SYS | 0x0100 |
| MT_MAC | 0x0200 |
| MT_NWK | 0x0300 |
| MT_AF | 0x0400 |
| MT_ZDO | 0x0500 |
| MT_SAPI | 0x0600 |
| MT_UTIL | 0x0700 |
| MT_DEBUG | 0x0800 |
| MT_APP | 0x0900 |
| ALL SUBSYSTEM | 0xFFFF |

SubsystemId — 2 (alongside the table above)

| Attribute | Length (byte) | Description |
|---|---|---|
| Action | 1 | 0: Disable, 1: Enable |

**SRSP:**

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x67 | Cmd1 = 0x06 | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.10.1.8   UTIL_KEY_EVENT

**Description:**

Sends key and shift codes to the application that registered for key events. The keys parameter is a bit mask, allowing for multiple keys in a single command. The return status indicates success if the command is processed by a registered key handler, not whether the key code was used. Not all applications support all key or shift codes but there is no indication when a key code is dropped.

**Usage:**

**SREQ:**

| 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|
| Length = 0x02 | Cmd0 = 0x27 | Cmd1 = 0x07 | Keys | Shift |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Keys | 1 | Key code bit mask: |
| | | 0x01: key 1 |
| | | 0x02: key 2 |
| | | 0x04: key 3 |
| | | 0x08: key 4 |
| | | 0x10: key 5 |
| | | 0x20: key 6 |
| | | 0x40: key 7 |
| | | 0x80: key 8 |
| Shift | 1 | 0: No shift, 1: Shift |

**SRSP:**

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x67 | Cmd1 = 0x07 | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.10.1.9    UTIL_TIME_ALIVE

**Description:**

This command is used by the tester to get the board's time alive.

**Usage:**

**SREQ:**

| 1 | 1 | 1 |
|---|---|---|
| Length = 0x00 | Cmd0 = 0x27 | Cmd1 = 0x09 |

**Attributes:**

None

**SRSP:**

| 1 | 1 | 1 | 4 |
|---|---|---|---|
| Length = 0x04 | Cmd0 = 0x67 | Cmd1 = 0x09 | Seconds |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Seconds | 4 | The time of the board's live in seconds |

## 3.10.1.10  UTIL_LED_CONTROL

**Description:**

This command is used by the tester to control the LEDs on the board.

**Usage:**

**SREQ:**

| 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|
| Length = 0x02 | Cmd0 = 0x27 | Cmd1 = 0x0A | LedId | Mode |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Laded | 1 | The LED number |
| Mode | 1 | 0: OFF, 1: ON |

**SRSP:**

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x67 | Cmd1 = 0x0A | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.10.1.11  UTIL_LOOPBACK

**Description:**

This command is used by the tester to test data buffer loopback.

**Usage:**

**SREQ:**

| 1 | 1 | 1 | 0 - 250 |
|---|---|---|---------|
| Length = 0x00 – 0xFA | Cmd0 = 0x27 | Cmd1 = 0x10 | Data |

**Attributes:**

| Attribute | Length (byte) | Description |
|-----------|---------------|-------------|
| Data | 0 – 250 | The data bytes to be looped back. |

**SRSP:**

| 1 | 1 | 1 | 0 - 250 |
|---|---|---|---------|
| Length = 0x00 – 0xFA | Cmd0 = 0x67 | Cmd1 = 0x10 | Data |

**Attributes:**

| Attribute | Length (byte) | Description |
|-----------|---------------|-------------|
| Data | 0 – 250 | The looped back data bytes. |

### 3.10.1.12  UTIL_DATA_REQ

**Description:**

This command is used by the tester to effect a MAC MLME Poll Request.

**Usage:**

**SREQ:**

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x27 | Cmd1 = 0x11 | SecurityUse |

**Attributes:**

| Attribute | Length | Description |
|-----------|--------|-------------|
| SecurityUse | 1 | TRUE to request MAC security, but not used for now. |

**SRSP:**

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x67 | Cmd1 = 0x11 | Status |

**Attributes:**

| Attribute | Length | Description |
|---|---|---|
| Status | 1 | A MAC status value from ZComDef.h, but only ZMacSuccess for now. |

### 3.10.1.13   UTIL_SRC_MATCH_ENABLE

**Description:**

This command is used to enable AUTOPEND and source address matching.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 |
|---|---|---|
| Length = 0x00 | Cmd0 = 0x27 | Cmd1 = 0x20 |

**SRSP**:

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x67 | Cmd1 = 0x20 | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.10.1.14   UTIL_SRC_MATCH_ADD_ENTRY

**Description:**

This command is used to add a short or extended address to the source address table.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 1 | 8 |
|---|---|---|---|---|
| Length = 0x0B | Cmd0 = 0x27 | Cmd1 = 0x21 | AddressMode | Address |

**Attributes**:

| Attribute | Length (byte) | Description | | |
|---|---|---|---|---|
| | | Mode | Value | Description |
| AddressMode | 1 | ADDRESS_16_BIT | 0x02 | Address 16 bit |
| | | ADDRESS_64_BIT | 0x03 | Address 64 bit |
| Address | 8 | Address of the device that will be added - Can be short or extended depends on the address mode. | | |
| PanId | 2 | PAN Id of the device. Only use when the address is a short address. | | |

**SRSP**:

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x67 | Cmd1 = 0x21 | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

## 3.10.1.15   UTIL_SRC_MATCH_DEL_ENTRY

**Description:**

This command is used to delete a short or extended address from the source address table.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 1 | 8 | 2 |
|---|---|---|---|---|---|
| Length = 0x0B | Cmd0 = 0x27 | Cmd1 = 0x22 | AddressMode | Address | PanId |

**Attributes**:

| Attribute | Length (byte) | Description | | |
|-----------|---------------|-------------|--|--|
| | | Mode | Value | Description |
| AddressMode | 1 | ADDRESS_16_BIT | 0x02 | Address 16 bit |
| | | ADDRESS_64_BIT | 0x03 | Address 64 bit |
| Address | 8 | Address of the device that will be deleted - Can be short or extended depends on the address mode. | | |
| PanId | 2 | PAN Id of the device. Only use when the address is a short address. | | |

**SRSP**:

| Byte: 1 | 1 | 1 | 1 |
|---------|---|---|---|
| Length = 0x01 | Cmd0 = 0x67 | Cmd1 = 0x22 | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|-----------|---------------|-------------|
| Status | 1 | Status is either Success (0) or Failure (1). |

## 3.10.1.16   UTIL_SRC_MATCH_CHECK_SRC_ADDR

**Description:**

This command is used to check if a short or extended address is in the source address table.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 1 | 8 | 2 |
|---------|---|---|---|---|---|
| Length = 0x0B | Cmd0 = 0x27 | Cmd1 = 0x23 | AddressMode | Address | PanId |

**Attributes**:

| Attribute | Length (byte) | Description | | |
|---|---|---|---|---|
| AddressMode | 1 | **Mode** | **Value** | **Description** |
| | | ADDRESS_16_BIT | 0x02 | Address 16 bit |
| | | ADDRESS_64_BIT | 0x03 | Address 64 bit |
| Address | 8 | Address of the device that will be checked - Can be short or extended depends on the address mode. | | |
| PanId | 2 | PAN Id of the device. Only use when the address is a short address. | | |

**SRSP**:

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x67 | Cmd1 = 0x23 | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

## 3.10.1.17  UTIL_SRC_MATCH_ACK_ALL_PENDING

**Description:**

This command is used to enable/disable acknowledging all packets with pending bit set.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x27 | Cmd1 = 0x24 | Option |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Option | 1 | TRUE - acknowledging all packets with pending field set. FALSE - Otherwise |

**SRSP**:

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x67 | Cmd1 = 0x24 | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.10.1.18  UTIL_SRC_MATCH_CHECK_ALL_PENDING

**Description:**

This command is used to check if acknowledging all packets with pending bit set is enabled.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 |
|---|---|---|
| Length = 0x00 | Cmd0 = 0x27 | Cmd1 = 0x25 |

**Attributes**:  None

**SRSP**:

| Byte: 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x67 | Cmd1 = 0x25 | Status | Value |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |
| Value | 1 | TRUE - acknowledging all packets with pending bit set is enabled. FALSE – otherwise. |

### 3.10.1.19  UTIL_ADDRMGR_EXT_ADDR_LOOKUP

**Description:**

This command is a proxy call to the AddrMgrEntryLookupExt() function.

**Usage:**

**SREQ:**

| 1 | 1 | 1 | 8 |
|---|---|---|---|
| Length = 0x08 | Cmd0 = 0x27 | Cmd1 = 0x40 | ExtAddr |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| ExtAddr | 8 | Extended Address (LSB-MSB) of the device for which to lookup the Network Address. |

**SRSP:**

| 1 | 1 | 1 | 2 |
|---|---|---|---|
| Length = 0x02 | Cmd0 = 0x67 | Cmd1 = 0x40 | NwkAddr |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| NwkAddr | 2 | Network Address (LSB-MSB) of the device that corresponds to the Extended Address sent as a parameter in the request. |

### 3.10.1.20   UTIL_ADDRMGR_NWK_ADDR_LOOKUP

**Description:**

This command is a proxy call to the AddrMgrEntryLookupNwk() function.

**Usage:**

**SREQ:**

| 1 | 1 | 1 | 2 |
|---|---|---|---|
| Length = 0x02 | Cmd0 = 0x27 | Cmd1 = 0x41 | NwkAddr |

**Attributes:**

| Attribute | Length (byte) | Description |
|-----------|---------------|-------------|
| NwkAddr | 2 | Network Address (LSB-MSB) of the device for which to lookup the Extended Address. |

**SRSP:**

| 1 | 1 | 1 | 8 |
|---|---|---|---|
| Length = 0x08 | Cmd0 = 0x67 | Cmd1 = 0x41 | ExtAddr |

**Attributes:**

| Attribute | Length (byte) | Description |
|-----------|---------------|-------------|
| ExtAddr | 8 | Extended Address (LSB-MSB) of the device that corresponds to the Network Address sent as a parameter in the request. |

### 3.10.1.21  UTIL_APSME_LINK_KEY_DATA_GET

**Description:**

This command retrieves APS link key data, Tx and Rx frame counters.

**Usage:**

**SREQ:**

| 1 | 1 | 1 | 8 |
|---|---|---|---|
| Length = 0x08 | Cmd0 = 0x27 | Cmd1 = 0x44 | ExtAddr |

**Attributes:**

| Attribute | Length (byte) | Description |
|-----------|---------------|-------------|
| ExtAddr | 8 | The extended address for which to get the link key data. |

**SRSP:**

| 1 | 1 | 1 | 1 | 16 | 4 | 4 |
|---|---|---|---|----|----|----|
| Length = 0x19 | Cmd0 = 0x67 | Cmd1 = 0x44 | Status | SecKey | TxFrmCntr | RxFrmCntr |

**Attributes:**

| Attribute | Length (byte) | Description |
|-----------|---------------|-------------|
| Status | 1 | The ZStatus_t returned by the proxy call to APSME_LinkKeyNVIdGet(). |
| SecKey | 16 | On success, the security key looked up; otherwise N/A. |
| TxFrmCntr | 4 | On success, the TX frame counter; otherwise N/A. |
| RxFrmCntr | 4 | On success, the RX frame counter, otherwise N/A. |

## 3.10.1.22   UTIL_APSME_LINK_KEY_NV_ID_GET

**Description:**

This command is a proxy call to the APSME_LinkKeyNvIdGet() function.

**Usage:**

**SREQ:**

| 1 | 1 | 1 | 8 |
|---|---|---|---|
| Length = 0x08 | Cmd0 = 0x27 | Cmd1 = 0x45 | ExtAddr |

**Attributes:**

| Attribute | Length (byte) | Description |
|-----------|---------------|-------------|
| ExtAddr | 8 | The extended address for which to get the link key NV Id. |

**SRSP:**

| 1 | 1 | 1 | 1 | 2 |
|---|---|---|---|---|
| Length = 0x03 | Cmd0 = 0x67 | Cmd1 = 0x45 | Status | LinkKeyNvId |

**Attributes:**

| Attribute | Length (byte) | Description |
|-----------|---------------|-------------|
| Status | 1 | Status of proxy call to APSME_LinkKeyNvIdGet(). |
| LinkKeyNvId | 2 | On success, link key NV ID. Otherwise 0xFFFF |

### 3.10.1.23  UTIL_APSME_REQUEST_KEY_CMD

**Description:**

This command is used to send a request key to the Trust Center from an originator device who wants to exchange messages with a partner device.

**Usage:**

**SREQ:**

| 1 | 1 | 1 | 2 |
|---|---|---|---|
| Length = 0x02 | Cmd0 = 0x27 | Cmd1 = 0x4B | PartnerAddr |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| PartnerAddr | 8 | Specifies the extended address of the partner device the originator wants to exchange messages with. |

**SRSP:**

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x67 | Cmd1 = 0x4B | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.10.1.24  UTIL_ASSOC_COUNT

**Description:**

This command is a proxy call to the AssocCount() function.

**Usage:**

**SREQ:**

| 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|
| Length = 0x02 | Cmd0 = 0x27 | Cmd1 = 0x48 | StartRelation | EndRelation |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| StartRelation | 1 | A valid node relation from AssocList.h:<br><br>// Node Relations<br><br>#define PARENT          0<br><br>#define CHILD_RFD       1<br><br>#define CHILD_RFD_RX_IDLE  2<br><br>#define CHILD_FFD       3<br><br>#define CHILD_FFD_RX_IDLE  4<br><br>#define NEIGHBOR        5<br><br>#define OTHER           6<br><br>The node relation at which to start counting. |
| EndRelation | 1 | Same as StartRelation, but the node relation at which to stop counting. |

**SRSP:**

| 1 | 1 | 1 | 2 |
|---|---|---|---|
| Length = 0x02 | Cmd0 = 0x67 | Cmd1 = 0x48 | Count |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Count | 2 | The count returned by the proxy call to AssocCount(). |

## 3.10.1.25   UTIL_ASSOC_FIND_DEVICE

**Description:**

This command is a proxy call to the AssocFindDevice() function.

**Usage:**

**SREQ:**

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x27 | Cmd1 = 0x49 | Number |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Number | 1 | Nth active entry in the device list. |

**SRSP:**

| 1 | 1 | 1 | 18 |
|---|---|---|---|
| Length = 0x12 | Cmd0 = 0x67 | Cmd1 = 0x49 | Device |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Device | 18 | The packed (LSB-MSB) associated_devices_t structure returned by the proxy call to AssocFindDevice().The device short address is set to INVALID_NODE_ADDR to indicate failure. |

### 3.10.1.26  UTIL_ASSOC_GET_WITH_ADDRESS

**Description:**

This command is a proxy call to the AssocGetWithAddress() function.

**Usage:**

**SREQ:**

| 1 | 1 | 1 | 8 | 2 |
|---|---|---|---|---|
| Length = 0x0A | Cmd0 = 0x27 | Cmd1 = 0x4A | ExtAddr | NwkAddr |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| ExtAddr | 8 | The extended address to use for the lookup or all zeroes to use the NwkAddr for the lookup. |
| NwkAddr | 2 | Network Address (LSB-MSB) to use for the lookup if the ExtAddr is all zeroes. |

**SRSP:**

| 1 | 1 | 1 | 18 |
|---|---|---|---|
| Length = 0x12 | Cmd0 = 0x67 | Cmd1 = 0x4A | Device |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Device | 18 | The packed (LSB-MSB) associated_devices_t structure returned by the proxy call to AssocGetWithAddress (). The device short address is set to INVALID_NODE_ADDR to indicate failure. |

### 3.10.1.27   UTIL_BIND_ADD_ENTRY

**Description:**

This command is a proxy call to the bindAddEntry() function.

**Usage:**

**SREQ:**

| 1 | 1 | 1 | 1 | 8 |
|---|---|---|---|---|
| Length = 0x0D-0x13 | Cmd0 = 0x27 | Cmd1 = 0x4D | AddrMode | DstAddr |

| 1 | 1 | 2-8 |
|---|---|---|
| DstEndpoint | NumClusterIds | ClusterIds |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| AddrMode | 1 | The destination address mode according to the afAddrMode_t enumeration in AF.h. |
| DstAddr | 8 | If AddrMode is afAddr64Bit, the 8-byte extended destination address; otherwise the 2-byte network destination address. |
| DstEndpoint | 1 | Binding entry destination endpoint. |
| NumClusterIds | 1 | Specifies the number of IDs in the ClusterIds list. |
| ClusterIds | 2-8 | Contains the cluster IDs. |

**SRSP:**

| 1 | 1 | 1 | 14 |
|---|---|---|---|
| Length = 0x0E | Cmd0 = 0x67 | Cmd1 = 0x4D | BindEntry |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| BindEntry | 14 | The packed (LSB-MSB) BindingEntry_t structure returned by the proxy call to bindAddEntry().The dstIdx in the BindEntry is set to INVALID_NODE_ADDR to indicate failure. |

### 3.10.1.28   UTIL_ZCL_KEY_EST_INIT_EST

**Description:**

This command is a proxy call to zclGeneral_KeyEstablish_InitiateKeyEstablishment().

**Usage:**

**SREQ:**

| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 8 |
|---|---|---|---|---|---|---|---|
| Length = 0x0C | Cmd0 = 0x27 | Cmd1 = 0x80 | TaskId | SeqNum | EndPoint | AddrMode | Addr |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| TaskId | 1 | The OSAL Task Id making the request. |
| SeqNum | 1 | The sequence number of the request. |
| EndPoint | 1 | The endpoint on the partner. |
| AddrMode | 1 | The address mode to the partner according to the afAddrMode_t enumeration in AF.h. |
| Addr | 8 | If AddrMode is afAddr64Bit, the 8-byte extended address of the partner; otherwise the 2-byte network address of the partner. |

**SRSP:**

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x67 | Cmd1 = 0x80 | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | The ZStatus_t returned by the proxy call to zclGeneral_KeyEstablish_InitiateKeyEstablishment(). |

## 3.10.1.29 UTIL_ZCL_KEY_EST_SIGN

**Description:**

This command is a proxy call to zclGeneral_KeyEstablishment_ECDSASign().

**Usage:**

**SREQ:**

| 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|
| Length = 0x0C | Cmd0 = 0x27 | Cmd1 = 0x81 | InputLen | Input |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| InputLen | 1 | The length of the input data. |
| Input | InputLen | The input data. |

**SRSP:**

| 1 | 1 | 1 | 1 | 42 |
|---|---|---|---|---|
| Length = 0x2B | Cmd0 = 0x67 | Cmd1 = 0x81 | Status | Key |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | The ZStatus_t returned by the proxy call to zclGeneral_KeyEstablishment_ECDSASign (). |
| Key | 42 | The output key on success. |

### 3.10.1.30   UTIL_SRNG_GEN

**Description:**

This command is used to generate Secure Random Number.  It generates 1,000,000 bits in sets of 100 bytes. As in 100 bytes of secure random numbers are generated until 1,000,000 bits are generated. 100 bytes are generated 1250 times. So 1250 SRSPs are generated. MT_SRNG has to be defined to include this API.

**Usage:**

**SREQ:**

| 1 | 1 | 1 |
|---|---|---|
| Length = 0x00 | Cmd0 = 0x27 | Cmd1 = 0x4C |

**SRSP:**

| 1 | 1 | 1 | 0x64 |
|---|---|---|---|
| Length = 0x64 | Cmd0 = 0x67 | Cmd1 = 0x4C | Secure Random Numbers |

### 3.10.2   MT_UTIL Callbacks

#### 3.10.2.1    UTIL_SYNC_REQ

**Description:**

This is an asynchronous request/response handshake.

**Usage:**

**AREQ:**

| 1 | 1 | 1 |
|---|---|---|
| Length = 0x00 | Cmd0 = 0x47 | Cmd1 = 0xE0 |

#### 3.10.2.2    UTIL_ZCL_KEY_ESTABLISH_IND

**Description:**

This is the RPC proxy indication for a ZCL_KEY_ESTABLISH_IND.

**Usage:**

**AREQ:**

| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 |
|---|---|---|---|---|---|---|---|
| Length = 0x06 | Cmd0 = 0x47 | Cmd1 = 0xE1 | TaskId | Event | Status | WaitTime | Suite |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| TaskId | 1 | The OSAL Task Id registered to receive this indication (see UTIL_ZCL_KEY_EST_INIT_EST). |
| Event | 1 | The OSAL message event. |
| Status | 1 | The OSAL message status. |
| WaitTime | 1 | The wait time. |
| Suite | 2 | The key establishment suite. |

### 3.11    MT_VERSION

This interface contains information about the release version of the software. There is no direct command for tester to interact with this interface.

### 3.12    MT_ZDO

This interface allows the tester to issue commands to the ZDO layer in the target and receive responses.  Each of these messages has a corresponding message that is returned by the target.  The response message only indicates that the command message was received and executed. The result of the command execution will be conveyed to the tester via a callback message interface.

#### 3.12.1  MT_ZDO Commands

##### 3.12.1.1    ZDO_NWK_ADDR_REQ

**Description:**

This message will request the device to send a "Network Address Request".  This message sends a broadcast message looking for a 16 bit address with a known 64 bit IEEE address.  You must subscribe to "ZDO Network Address Response" to receive the response to this message. Check section 3.0.1.7 for more details on callback subscription. The response message listed below only indicates whether or not the message was received properly.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 8 | 1 | 1 |
|---|---|---|---|---|---|
| Length = 0x0A | Cmd0 = 0x25 | Cmd1 = 0x00 | IEEEAddress | ReqType | StartIndex |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| IEEEAddress | 8 | 64 bit IEEE address of the device. |
| | | Value that the search was executed on. |
| ReqType | 1 | <table><tr><td>Type</td><td>Value</td></tr><tr><td>Single Device response</td><td>0x00</td></tr><tr><td>Extended, include associated devices</td><td>0x01</td></tr></table> |
| StartIndex | 1 | Starting index into the list of children.  This is used to get more of the list if the list is too large for one message. |

**SRSP:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x65 | Cmd1 = 0x00 | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|-----------|---------------|-------------|
| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.12.1.2   ZDO_IEEE_ADDR_REQ

**Description:**

This command will request a device's IEEE 64-bit address.  You must subscribe to "ZDO IEEE Address Response" to receive the data response to this message.  The response message listed below only indicates whether or not the message was received properly.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 2 | 1 | 1 |
|---------|---|---|---|---|---|
| Length = 0x04 | Cmd0 = 0x25 | Cmd1 = 0x01 | ShortAddr | ReqType | StartIndex |

**Attributes**:

| Attribute | Length (byte) | Description |
|-----------|---------------|-------------|
| ShortAddr | 2 | Specifies the short address of the device. Value that the search was executed on. |
| ReqType | 1 | <table><tr><td>Type</td><td>Value</td></tr><tr><td>Single Device response</td><td>0x00</td></tr><tr><td>Extended, include associated devices</td><td>0x01</td></tr></table> |
| StartIndex | 1 | Starting index into the list of children.  This is used to get more of the list if the list is too large for one message. |

**SRSP:**

| Byte: 1 | 1 | 1 | 1 |
|---------|---|---|---|
| Length = 0x01 | Cmd0 = 0x65 | Cmd1 = 0x01 | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|-----------|---------------|-------------|
| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.12.1.3   ZDO_NODE_DESC_REQ

**Description:**

This command is generated to inquire about the Node Descriptor information of the destination device.

**Usage**

**SREQ:**

| 1 | 1 | 1 | 2 | 2 |
|---|---|---|---|---|
| Length = 0x04 | Cmd0 = 0x25 | Cmd1 = 0x02 | DstAddr | NWKAddrOfInterest |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| DstAddr | 2 | Specifies NWK address of the device to route the inquiry. |
| NWKAddrOfInterest | 2 | Specifies NWK address of the destination device being queried. |

**SRSP:**

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x65 | Cmd1 = 0x02 | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.12.1.4   ZDO_POWER_DESC_REQ

**Description:**

This command is generated to inquire about the Power Descriptor information of the destination device.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 2 | 2 |
|---|---|---|---|---|
| Length = 0x04 | Cmd0 = 0x25 | Cmd1 = 0x03 | DstAddr | NWKAddrOfInterest |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| DstAddr | 2 | Specifies NWK address of the device to route the inquiry. |
| NWKAddrOfInterest | 2 | Specifies NWK address of the destination device being queried. |

**SRSP:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x65 | Cmd1 = 0x03 | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

## 3.12.1.5    ZDO_SIMPLE_DESC_REQ

**Description:**

This command is generated to inquire as to the Simple Descriptor of the destination device's Endpoint.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 2 | 2 | 1 |
|---|---|---|---|---|---|
| Length = 0x05 | Cmd0 = 0x25 | Cmd1 = 0x04 | DstAddr | NWKAddrOfInterest | Endpoint |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| DstAddr | 2 | Specifies NWK address of the device to route the inquiry. |
| NWKAddrOfInterest | 2 | Specifies NWK address of the destination device being queried. |
| Endpoint | 1 | Specifies the application endpoint the data is from. |

**SRSP:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x65 | Cmd1 = 0x04 | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|-----------|---------------|-------------|
| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.12.1.6   ZDO_ACTIVE_EP_REQ

**Description:**

This command is generated to request a list of active endpoint from the destination device.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 2 | 2 |
|---------|---|---|---|---|
| Length = 0x04 | Cmd0 = 0x25 | Cmd1 = 0x05 | DstAddr | NWKAddrOfInterest |

**Attributes**:

| Attribute | Length (byte) | Description |
|-----------|---------------|-------------|
| DstAddr | 2 | Specifies NWK address of the device to route the inquiry. |
| NWKAddrOfInterest | 2 | Specifies NWK address of the destination device being queried. |

**SRSP:**

| Byte: 1 | 1 | 1 | 1 |
|---------|---|---|---|
| Length = 0x01 | Cmd0 = 0x65 | Cmd1 = 0x05 | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|-----------|---------------|-------------|
| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.12.1.7   ZDO_MATCH_DESC_REQ

**Description:**

This command is generated to request the device match descriptor.

**Usage:**

**SREQ:**

| 1 | 1 | 1 | 2 | 2 | 2 |
|---|---|---|---|---|---|
| Length = 0x08-0x48 | Cmd0 = 0x25 | Cmd1 = 0x06 | DstAddr | NwkAddrOfInterest | ProfileId |

| 1 | 0-32 | 1 | 0-32 |
|---|---|---|---|
| NumInClusters | InClusterList | NumOutClusters | OutClusterList |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| DstAddr | 2 | Specifies NWK address of the device to route the inquiry. |
| NWKAddrOfInterest | 2 | Specifies NWK address of the destination device being queried. |
| ProfileId | 2 | Specifies the profile Id of the device |
| NumInClusters | 1 | Specifies the number of Id's in the InClusterList. |
| InClusterList | 0-32 | Contains the input cluster Id's. |
| NumOutClusters | 1 | Specifies the number of Id's in the OutClusterList. |
| OutClusterList | 0-32 | Contains the output cluster Id's. |

**SRSP:**

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x65 | Cmd1 = 0x06 | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.12.1.8    ZDO_COMPLEX_DESC_REQ

**Description:**

This command is generated to request for the destination device's complex descriptor.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 2 | 2 |
|---|---|---|---|---|
| Length = 0x04 | Cmd0 = 0x25 | Cmd1 = 0x07 | DstAddr | NWKAddrOfInterest |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| DstAddr | 2 | Specifies NWK address of the device to route the inquiry. |
| NWKAddrOfInterest | 2 | Specifies NWK address of the destination device being queried. |

**SRSP:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x65 | Cmd1 = 0x07 | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.12.1.9    ZDO_USER_DESC_REQ

**Description:**

This command is generated to request for the destination device's user descriptor.

**Usage:**

**SREQ:**

| 1 | 1 | 1 | 2 | 2 |
|---|---|---|---|---|
| Length = 0x04 | Cmd0 = 0x25 | Cmd1 = 0x08 | DstAddr | NWKAddrOfInterest |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|

| DstAddr | 2 | Specifies NWK address of the device to process the inquiry. |
| NWKAddrOfInterest | 2 | Specifies NWK address of the destination device being queried. |

**SRSP:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x65 | Cmd1 = 0x08 | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

## 3.12.1.10   ZDO_END_DEVICE_ANNCE

**Description:**

This command will cause the device to issue an "End device announce" broadcast packet to the network. This is typically used by an end-device to announce itself to the network.

**Usage:**

**SREQ:**

| 1 | 1 | 1 | 2 | 8 | 1 |
|---|---|---|---|---|---|
| Length = 0x0B | Cmd0 = 0x25 | Cmd1 = 0x0A | NwkAddr | IEEEAddr | Capabilites |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| NwkAddr | 2 | Specifies network address of the device generating the request. |
| IEEEAddr | 8 | Specifies the 64 bit IEEE Address of the device being announced. |
| Capabilites | 1 | Specifies MAC capabilities<br><br>Bit: 0 – Alternate PAN Coordinator<br><br>1 – Device type: 1- ZigBee Router; 0 – End Device<br><br>2 – Power Source: 1 Main powered<br><br>3 – Receiver on when Idle<br><br>4 – Reserved<br><br>5 – Reserved<br><br>6 – Security capability |

7 – Reserved

**SRSP:**

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x65 | Cmd1 = 0x0A | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.12.1.11  ZDO_USER_DESC_SET

**Description:**

This command is generated to write a User Descriptor value to the targeted device.

**Usage:**

**SREQ:**

| 1 | 1 | 1 | 2 | 2 | 1 | 0-16 |
|---|---|---|---|---|---|---|
| Length = 0x05-0x15 | Cmd0 = 0x25 | Cmd1 = 0x0B | DstAddr | NWKAddrOfInterest | Len | UserDescriptor |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| DstAddr | 2 | Specifies network address of the device to route the set request. |
| NWKAddrOfInterest | 2 | Specifies NWK address of the destination device being queried. |
| Len | 1 | Specifies the length of the user descriptor. |
| UserDescriptor | 0-16 | User descriptor array (can be up to 16 bytes). |

**SRSP:**

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x65 | Cmd1 = 0x0B | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.12.1.12  ZDO_SERVER_DISC_REQ

**Description:**

The command is used for local device 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 RxOnWhenIdle devices'.

**Usage:**

**SREQ:**

| 1 | 1 | 1 | 2 |
|---|---|---|---|
| Length = 0x02 | Cmd0 = 0x25 | Cmd1 = 0x0C | ServerMask |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| ServerMask | 2 | Specifies the system server capabilities of the device. |

**SRSP:**

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x65 | Cmd1 = 0x0C | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.12.1.13  ZDO_END_DEVICE_BIND_REQ

**Description:**

This command is generated to request an End Device Bind with the destination device.

**Usage:**

**SREQ:**

| 1 | 1 | 1 | 2 | 2 | 1 |
|---|---|---|---|---|---|
| Length = 0x09-0x49 | Cmd0 = 0x25 | Cmd1 = 0x20 | DstAddr | LocalCoordinator | Endpoint |

| 2 | 1 | 0-32 | 1 | 0-32 |
|---|---|---|---|---|
| ProfileId | NumInClusters | InClusterList | NumOutClusters | OutClusterList |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| DstAddr | 2 | Specifies NWK address of the target device for binding. |
| LocalCoordinator | 2 | Specifies Specifies local coordinator's short address. In the case of source binding, it's the short address of the source address |
| IEEE | 8 | Local coordinator's IEEE address |
| Endpoint | 1 | Device's endpoint. |
| ProfileId | 2 | Specifies the profile Id of the device. |
| NumInClusters | 1 | Specifies the number of Id's in the InClusterList. |
| InClusterList | 0-32 | Contains the input cluster Id's. |
| NumOutClusters | 1 | Specifies the number of Id's in the OutClusterList. |
| OutClusterList | 0-32 | Contains the output cluster Id's. |

**SRSP:**

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x65 | Cmd1 = 0x20 | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.12.1.14  ZDO_BIND_REQ

**Description:**

This command is generated to request a Bind.

**Usage:**

**SREQ:**

| 1 | 1 | 1 | 2 | 8 | 1 | 2 |
|---|---|---|---|---|---|---|
| | | | | | | |

| Length = 0x10-0x17 | Cmd0 = 0x25 | Cmd1 = 0x21 | DstAddr | SrcAddress | SrcEndpoint | ClusterId |
|---|---|---|---|---|---|---|

| 1 | 2/8 | 0/1 |
|---|---|---|
| DstAddrMode | DstAddress | DstEndpoint |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| DstAddr | 2 | Specifies the destination address of the device to process the bind request |
| SrcAddress | 8 | 64 bit Binding source IEEE address |
| SrcEndpoint | 1 | Specifies the binding source endpoint. |
| ClusterId | 2 | Specifies the cluster Id to match in messages. |
| DstAddrMode | 1 | Specifies destination address mode<br><br><table><tr><td>Mode</td><td>Value</td><td>Description</td></tr><tr><td>ADDRESS_NOT_PRESENT</td><td>0x00</td><td>Address Not Present</td></tr><tr><td>GROUP_ADDRESS</td><td>0x01</td><td>Group address</td></tr><tr><td>ADDRESS_16_BIT</td><td>0x02</td><td>Address 16 bit</td></tr><tr><td>ADDRESS_64_BIT</td><td>0x03</td><td>Address 64 bit</td></tr><tr><td>BROADCAST</td><td>0xFF</td><td>Broadcast</td></tr></table> |
| DstAddress | 8/2 | Binding destination IEEE address. Not to be confused with DstAddr. |
| DstEndpoint | 1/0 | Specifies the binding destination endpoint. It is used only when DstAddrMode is 64 bits extended address |

**SRSP:**

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x65 | Cmd1 = 0x21 | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.12.1.15   ZDO_UNBIND_REQ

**Description:**

This command is generated to request a un-bind.

**Usage:**

**SREQ:**

| 1 | 1 | 1 | 2 | 8 | 1 | 2 |
|---|---|---|---|---|---|---|
| Length = 0x10-0x17 | Cmd0 = 0x25 | Cmd1 = 0x22 | DstAddr | SrcAddress | SrcEndpoint | ClusterId |

| 1 | 2/8 | 0/1 |
|---|-----|-----|
| DstAddrMode | DstAddress | DstEndpoint |

**Attributes:**

| Attribute | Length (byte) | Description |
|-----------|---------------|-------------|
| DstAddr | 2 | Specifies destination address of the device to process the unbind request. |
| SrcAddress | 8 | Specifies 64 bit Binding source IEEE address. |
| SrcEndpoint | 1 | Specifies the binding source endpoint. |
| ClusterI | 2 | Specifies cluster Id to match in messages. |
| DstAddrMode | 1 | Specifies 64 bit Binding destination address mode: <table><tr><td>Mode</td><td>Value</td><td>Description</td></tr><tr><td>ADDRESS_NOT_PRESENT</td><td>0x00</td><td>Address Not Present</td></tr><tr><td>GROUP_ADDRESS</td><td>0x01</td><td>Group address</td></tr><tr><td>ADDRESS_16_BIT</td><td>0x02</td><td>Address 16 bit</td></tr><tr><td>ADDRESS_64_BIT</td><td>0x03</td><td>Address 64 bit</td></tr><tr><td>BROADCAST</td><td>0xFF</td><td>Broadcast</td></tr></table> |
| DstAddress | 8 | Specifies 64 bit Binding destination IEEE address. Not to be confused with DstAddr. |
| DstEndpoint | 1 | Specifies the binding destination endpoint |

**SRSP:**

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x65 | Cmd1 = 0x22 | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.12.1.16  ZDO_MGMT_NWK_DISC_REQ

**Description:**

This command is generated to request the destination device to perform a network discovery.

**Usage:**

**SREQ:**

| 1 | 1 | 1 | 2 | 4 | 1 | 1 |
|---|---|---|---|---|---|---|
| Length = 0x08 | Cmd0 = 0x25 | Cmd1 = 0x30 | DstAddr | ScanChannels | ScanDuration | StartIndex |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| DstAddr | 2 | Specifies the network address of the device performing the discovery. |
| ScanChannels | 4 | Specifies the Bit Mask for channels to scan: <table><tr><td>Channel</td><td>Value</td></tr><tr><td>NONE</td><td>0x00000000</td></tr><tr><td>ALL_CHANNELS</td><td>0x07FFF800</td></tr><tr><td>CHANNEL 11</td><td>0x00000800</td></tr><tr><td>CHANNEL 12</td><td>0x00001000</td></tr><tr><td>CHANNEL 13</td><td>0x00002000</td></tr><tr><td>CHANNEL 14</td><td>0x00004000</td></tr><tr><td>CHANNEL 15</td><td>0x00008000</td></tr><tr><td>CHANNEL 16</td><td>0x00010000</td></tr><tr><td>CHANNEL 17</td><td>0x00020000</td></tr><tr><td>CHANNEL 18</td><td>0x00040000</td></tr><tr><td>CHANNEL 19</td><td>0x00080000</td></tr></table> |

| CHANNEL 20 | 0x00100000 |
|------------|------------|
| CHANNEL 21 | 0x00200000 |
| CHANNEL 22 | 0x00400000 |
| CHANNEL 23 | 0x00800000 |
| CHANNEL 24 | 0x01000000 |
| CHANNEL 25 | 0x02000000 |
| CHANNEL 26 | 0x04000000 |

| | | |
|---|---|---|
| ScanDuration | 1 | Specifies the scanning time. |
| StartIndex | 1 | Specifies where to start in the response array list.  The result may contain more entries than can be reported, so this field allows the user to retrieve the responses anywhere in the array list. |

**SRSP:**

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x65 | Cmd1 = 0x30 | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|-----------|---------------|-------------|
| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.12.1.17  ZDO_MGMT_LQI_REQ

**Description:**

This command is generated to request the destination device to perform a LQI query of other devices in the network.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 2 | 1 |
|---------|---|---|---|---|
| Length = 0x03 | Cmd0 = 0x25 | Cmd1 = 0x31 | DstAddr | StartIndex |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| DstAddr | 2 | Specifies the network address of the device to process the query. |
| StartIndex | 1 | Specifies where to start in the response array list.  The result may contain more entries than can be reported, so this field allows the user to retrieve the responses anywhere in the array list. |

**SRSP:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x65 | Cmd1 = 0x31 | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

## 3.12.1.18   ZDO_MGMT_RTG_REQ

**Description:**

This command is generated to request the Routing Table of the destination device.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 2 | 1 |
|---|---|---|---|---|
| Length = 0x03 | Cmd0 = 0x25 | Cmd1 = 0x32 | DstAddr | StartIndex |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| DstAddr | 2 | Specifies the network address of the device to process the query. |
| StartIndex | 1 | Specifies where to start in the response array list.  The result may contain more entries than can be reported, so this field allows the user to retrieve the responses anywhere in the array list. |

**SRSP:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x65 | Cmd1 = 0x32 | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|-----------|---------------|-------------|
| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.12.1.19   ZDO_MGMT_BIND_REQ

**Description**

This command is generated to request the Binding Table of the destination device.

**Usage**

**SREQ:**

| Byte: 1 | 1 | 1 | 2 | 1 |
|---------|---|---|---|---|
| Length = 0x03 | Cmd0 = 0x25 | Cmd1 = 0x33 | DstAddr | StartIndex |

**Attributes:**

| Attribute | Length (byte) | Description |
|-----------|---------------|-------------|
| DstAddr | 2 | Specifies the network address of the device being queried. |
| StartIndex | 1 | Specifies where to start in the response array list.  The result may contain more entries than can be reported, so this field allows the user to retrieve the responses anywhere in the array list. |

**SRSP:**

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x65 | Cmd1 = 0x33 | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|-----------|---------------|-------------|
| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.12.1.20  ZDO_MGMT_LEAVE_REQ

**Description:**

This command is generated to request a Management Leave Request for the target device.

**Usage:**

**SREQ:**

| 1 | 1 | 1 | 2 | 8 | 1 |
|---|---|---|---|---|---|
| Length = 0x0B | Cmd0 = 0x25 | Cmd1 = 0x34 | DstAddr | DeviceAddr | RemoveChildren_Rejoin |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| DstAddr | 2 | Short address of the device that will process the mgmt leave (remote or self) |
| DeviceAddress | 8 | The 64-bit IEEE address of the entity to be removed from the network or 0x0000000000000000 if the device removes itself from the network. |
| RemoveChildren_Rejoin | 1 | Specifies actions to be performed by device when leaving the network:  Bit 0:  rejoin    Bit 1:  remove children |

**SRSP:**

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x65 | Cmd1 = 0x34 | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.12.1.21  ZDO_MGMT_DIRECT_JOIN_REQ

**Description:**

This command is generated to request the Management Direct Join Request of a designated device.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 2 | 8 | 1 |
|---|---|---|---|---|---|
| Length = 0x0B | Cmd0 = 0x25 | Cmd1 = 0x35 | DstAddr | DeviceAddr | CapInfo |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| DstAddr | 2 | Network address of the device to which the device specified in DeviceAddress is to join. |
| DeviceAddress | 8 | The 64 bit IEEE Address of the device you want to be joined to the device at DstAddr. |
| CapInfo | 1 | Specifies the operating capabilities of the device being directly joined.  Bit weighted values follow:<br><br>Bit: 0 – Alternate PAN Coordinator<br><br>1 – Device type: 1- ZigBee Router; 0 – End Device<br><br>2 – Power Source: 1 Main powered<br><br>3 – Receiver on when Idle<br><br>4 – Reserved<br><br>5 – Reserved<br><br>6 – Security capability<br><br>7 – Reserved |

**SRSP:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x65 | Cmd1 = 0x35 | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.12.1.22  ZDO_MGMT_PERMIT_JOIN_REQ

**Description:**

This command is generated to set the Permit Join for the destination device.

**Usage:**

**SREQ:**

| 1 | 1 | 1 | 1 | 2 | 1 | 1 |
|---|---|---|---|---|---|---|
| Length = 0x05 | Cmd0 = 0x25 | Cmd1 = 0x36 | AddrMode | DstAddr | Duration | TCSignificance |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| AddrMode | | Destination address type: 0x02 – Address 16 bit, 0xFF – Broadcast. |
| DstAddr | 2 | Specifies the network address of the destination device whose Permit Join information is to be modified. |
| Duration | 1 | Specifies the duration to permit joining.  0 = join disabled.  0xff = join enabled. 0x01-0xfe = number of seconds to permit joining. |
| TCSignificance | 1 | Trust Center Significance. |

**SRSP:**

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x65 | Cmd1 = 0x36 | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.12.1.23  ZDO_MGMT_NWK_UPDATE_REQ

**Description:**

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.

**Usage:**

**SREQ:**
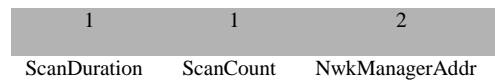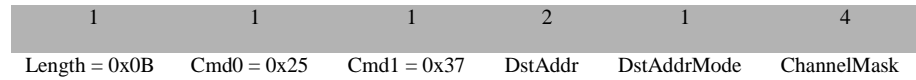
| 1 | 1 | 1 | 2 | 1 | 4 |
|---|---|---|---|---|---|
| Length = 0x0B | Cmd0 = 0x25 | Cmd1 = 0x37 | DstAddr | DstAddrMode | ChannelMask |

| 1 | 1 | 2 |
|---|---|---|
| ScanDuration | ScanCount | NwkManagerAddr |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| DstAddr | 2 | Short address of the destination device(s). The destination addressing on this primitive can be unicast or broadcast to all devices for which macRxOnWhenIdle=TRUE (i.e., 0xFFFD) |
| DstAddrMode | 1 | Destination address mode: |

| Mode | Value | Description |
|---|---|---|
| ADDRESS_NOT_PRESENT | 0x00 | Address Not Present |
| GROUP_ADDRESS | 0x01 | Group address |
| ADDRESS_16_BIT | 0x02 | Address 16 bit |
| ADDRESS_64_BIT | 0x03 | Address 64 bit |
| BROADCAST | 0xFF | Broadcast |

A bitmap indicating which channels are to be scanned:

| Channel | Value |
|---|---|
| NONE | 0x00000000 |
| ALL_CHANNELS | 0x07FFF800 |
| CHANNEL 11 | 0x00000800 |
| CHANNEL 12 | 0x00001000 |
| CHANNEL 13 | 0x00002000 |
| CHANNEL 14 | 0x00004000 |
| CHANNEL 15 | 0x00008000 |
| CHANNEL 16 | 0x00010000 |
| CHANNEL 17 | 0x00020000 |
| CHANNEL 18 | 0x00040000 |
| CHANNEL 19 | 0x00080000 |
| CHANNEL 20 | 0x00100000 |
| CHANNEL 21 | 0x00200000 |
| CHANNEL 22 | 0x00400000 |
| CHANNEL 23 | 0x00800000 |
| CHANNEL 24 | 0x01000000 |
| CHANNEL 25 | 0x02000000 |
| CHANNEL 26 | 0x04000000 |

ChannelMask — 4

| Field | Length | Description |
|---|---|---|
| ScanDuration | 1 | A value used to calculate the length of time to spend scanning each channel |
| ScanCount | 1 | This field represents the number of energy scans to be conducted and reported |
| NwkManagerAddr | 2 | Indicates the NWK address for the device with the Network Manager bit set in its Node Descriptor |

**SRSP:**
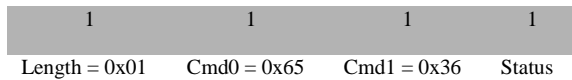
| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x65 | Cmd1 = 0x36 | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.12.1.24　ZDO_MSG_CB_REGISTER

**Description:**

This command registers for a ZDO callback (see reference [3], "6. ZDO Message Requests" for example usage).

**Usage:**

**SREQ:**

| 1 | 1 | 1 | 2 |
|---|---|---|---|
| Length = 0x02 | Cmd0 = 0x25 | Cmd1 = 0x3E | ClusterId |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| ClusterId | 2 | Specifies the ZDO Cluster Id for which to receive a ZDO callback. |

**SRSP:**

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x65 | Cmd1 = 0x3E | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Return value of the call to ZDO_RegisterForZDOMsg(). |

### 3.12.1.25  ZDO_MSG_CB_REMOVE

**Description:**

This command removes a registration for a ZDO callback (see reference [3], "6. ZDO Message Requests" for example usage).

**Usage:**

**SREQ:**

| 1 | 1 | 1 | 2 |
|---|---|---|---|
| Length = 0x02 | Cmd0 = 0x25 | Cmd1 = 0x3F | ClusterId |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| ClusterId | 2 | Specifies the ZDO Cluster Id for which to receive a ZDO callback. |

**SRSP:**

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x65 | Cmd1 = 0x3F | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Return value of the call to ZDO_RemoveRegisteredCB (). |

### 3.12.1.26  ZDO_STARTUP_FROM_APP

**Description:**

This command starts the device in the network.

**Usage:**

**SREQ:**

| 1 | 1 | 1 | 2 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x25 | Cmd1 = 0x40 | StartDelay |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| StartDelay | 2 | Specifies the time delay before the device starts. |

**SRSP:**

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x65 | Cmd1 = 0x40 | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | 0x00 – Restored  network state |
| | | 0x01 – New network state |
| | | 0x02 – Leave and not Started |

## 3.12.1.27   ZDO_STARTUP_FROM_APP_EX

**Description:**

Extended version of ZDO to indicate to router devices to create a distributed network.

**Usage:**

**SREQ:**

| 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x25 | Cmd1 = 0x54 | StartDelay | Mode |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| StartDelay | 1 | Specifies the time delay before the device starts. |
| Mode | 1 | Flag for ZR devices to indicate to create a distributed network if set to TRUE, if set to FALSE or send to ZC or ZED has the same effect as non-extended interface. |

**SRSP:**

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x65 | Cmd1 = 0x40 | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | 0x00 – Restored  network state |
| | | 0x01 – New network state |

### 3.12.1.28 ZDO_SET_LINK_KEY

**Description:**

This command sets the application link key for a given device.

**Usage:**

**SREQ:**

| 1 | 1 | 1 | 2 | 8 | 16 |
|---|---|---|---|---|---|
| Length = 0x1A | Cmd0 = 0x25 | Cmd1 = 0x23 | ShortAddr | IEEEAddr | LinkKeyData |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| ShortAddr | 2 | Specifies the short address of the pair device of the link key. |
| IEEEAddr | 8 | Specifies the IEEE address of the pair device of the link key |
| LinkKeyData | 16 | 128 bit link key data of the device. |

**SRSP:**

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x65 | Cmd1 = 0x23 | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | 0x00 – Success<br>0x01 – Fail to add to address manager.<br>0x11 – Security manager key table full |

### 3.12.1.29  ZDO_REMOVE_LINK_KEY

**Description:**
This command removes the application link key of a given device.

**Usage:**
**SREQ:**

| 1 | 1 | 1 | 8 |
|---|---|---|---|
| Length = 0x08 | Cmd0 = 0x25 | Cmd1 = 0x24 | IEEEAddr |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| IEEEAddr | 8 | Specifies the IEEE address of the pair device of the link key |

**SRSP:**

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x65 | Cmd1 = 0x24 | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | 0x00 – Success<br><br>0xC8 – Unknown device. |

### 3.12.1.30  ZDO_GET_LINK_KEY

**Description:**
This command retrieves the application link key of a given device.

**Usage:**
**SREQ:**

| 1 | 1 | 1 | 8 |
|---|---|---|---|
| Length = 0x08 | Cmd0 = 0x25 | Cmd1 = 0x25 | IEEEAddr |

**Attributes:**

| Attribute | Length (byte) | Description |
|-----------|---------------|-------------|
| IEEEAddr | 8 | Specifies the IEEE address of the pair device of the link key |

**SRSP:**

| 1 | 1 | 1 | 1 | 8 | 16 |
|---|---|---|---|---|----|
| Length = 0x19 | Cmd0 = 0x65 | Cmd1 = 0x25 | Status | IEEEAddr | LinkKeyData |

**Attributes:**

| Attribute | Length (byte) | Description |
|-----------|---------------|-------------|
| Status | 1 | 0x00 – Success <br><br> 0xC8 – Unknown device. |
| IEEEAddr | 8 | IEEE address of the device |
| LinkKeyData | 16 | Link key data of the device. |

## 3.12.1.31  ZDO_NWK_DISCOVERY_REQ

**Description:**

This command is used to initiate a network discovery (active scan).

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 4 | 1 |
|---------|---|---|---|---|
| Length = 0x05 | Cmd0 = 0x25 | Cmd1 = 0x26 | Scan Channels | Scan Duration |

**Attributes:**

| Attribute | Length (byte) | Description | | |
|-----------|---------------|-------------|---|---|
| | | Bit mask channels Type: | | for to scan. |
| | | | Channel | Value |
| Scan Channels | 4 | | | |
| | | ZIGBEE_CHANNELS | | |

| | |
|---|---|
| NONE | 0x00000000 |
| ALL_CHANNELS | 0x07FFF800 |
| CHANNEL 11 | 0x00000800 |
| CHANNEL 12 | 0x00001000 |
| CHANNEL 13 | 0x00002000 |
| CHANNEL 14 | 0x00004000 |
| CHANNEL 15 | 0x00008000 |
| CHANNEL 16 | 0x00010000 |
| CHANNEL 17 | 0x00020000 |
| CHANNEL 18 | 0x00040000 |
| CHANNEL 19 | 0x00080000 |
| CHANNEL 20 | 0x00100000 |
| CHANNEL 21 | 0x00200000 |
| CHANNEL 22 | 0x00400000 |
| CHANNEL 23 | 0x00800000 |
| CHANNEL 24 | 0x01000000 |
| CHANNEL 25 | 0x02000000 |
| CHANNEL 26 | 0x04000000 |

| | | |
|---|---|---|
| Scan Duration | 1 | A value used to calculate the length of time to spend scanning each channel |

**SRSP:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x65 | Cmd1 = 0x26 | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Success (0)<br><br>Invalid_Parameter (0x02).<br><br>ZNwkInvalidRequest (0xC2) if the device is already on a network. User ZDO_MGMT_NWK_DISC_REQ instead. Or leave the network first, then initiate the request.<br><br>MAC_SCAN_IN_PROGRESS (0xFC) if a channel change is in progress. |

MAC_NO_RESOURCE (0x1A) if the operation could not complete because no memory resource were available.

### 3.12.1.32  ZDO_JOIN_REQ

**Description**:

This command is used to request the device to join itself to a parent device on a network.

**Usage**:

**SREQ:**

| Byte: 1 | 1 | 1 | 1 | 2 |
|---------|---|---|---|---|
| Length = 0x0F | Cmd0 = 0x25 | Cmd1 = 0x27 | Logical Channel | Pan ID |

| 8 | 2 | 1 | 1 |
|---|---|---|---|
| Extended Pan ID | Chosen Parent | Parent Depth | Stack Profile |

**Attributes**:

| Attribute | Length (byte) | Description |
|-----------|--------------|-------------|
| Logical Channel | 1 | Channel where the PAN is located |
| Pan ID | 2 | Id of PAN to join |
| Extended Pan ID | 8 | 64-bit extended PAN ID (ver. 1.1 only). If not v1.1 or don't care, use all 0xFF |
| Chosen Parent | 2 | Short address of the parent device chosen to join |
| Parent Depth | 1 | Depth of the parent |
| Stack Profile | 1 | Stack profile of the network to join |

**SRSP:**

| Byte: 1 | 1 | 1 | 1 |
|---------|---|---|---|
| Length = 0x01 | Cmd0 = 0x65 | Cmd1 = 0x27 | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Success (0)<br><br>ZNwkInvalIdRequest (0xC2) if device is already on a network. Leave the network first, then try to join again.<br><br>ZNwkNotPermitted (0xC3) if chosen router is not a valid short address. |

### 3.12.1.33  ZDO_SET_REJOIN_PARAMETERS

**Description:**

This command is used to set rejoin backoff duration and rejoin scan duration for an end device

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 4 | 4 |
|---|---|---|---|---|
| Length = 0x08 | Cmd0 = 0x25 | Cmd1 = 0x26 | Backoff Duration | Scan Duration |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Backoff Duration | 4 | Rejoin backoff duration for end device |
| Scan Duration | 4 | Rejoin scan duration for end device |

**SRSP:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x65 | Cmd1 = 0x27 | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Success (0) |

### 3.12.1.34  ZDO_SEC_ADD_LINK_KEY

**Description:**

This command handles the ZDO security add link key extension message.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 2 | 8 | 16 |
|---|---|---|---|---|---|
| Length = 0x1A | Cmd0 = 0x25 | Cmd1 = 0x42 | Short Address | Extended Adress | Key |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Short Address | 2 | short address of the partner device |
| Extended Address | 8 | extended address of the partner device |
| Key | 16 | Link key |

**SRSP:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x65 | Cmd1 = 0x42 | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Success (0) or Failure(1) |

### 3.12.1.35   ZDO_SEC_ENTRY_LOOKUP_EXT
**Description:**

This command handles the ZDO security entry lookup extended extension message.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 8 | 5 |
|---|---|---|---|---|
| Length = 0x013 | Cmd0 = 0x25 | Cmd1 = 0x43 | Extended Adress | Entry |

**Attributes**:

| Attribute | Length (byte) | Description |
| --- | --- | --- |
| Extended Address | 8 | extended address |
| entry | 5 | Valid entry |

**SRSP:**

| Byte: 1 | 1 | 1 | 2 | 2 | 1 |
| --- | --- | --- | --- | --- | --- |
| Length = 0x05 | Cmd0 = 0x65 | Cmd1 = 0x42 | AMI | KeyNVID | Authentication option |

**Attributes**:

| Attribute | Length (byte) | Description |
| --- | --- | --- |
| AMI | 2 | Address manager index |
| KeyNVID | 2 | Index to link key table in NV |
| Authentication option | 1 | Authentication option for device |

### 3.12.1.36   ZDO_SEC_DEVICE_REMOVE

**Description:**

This command handles the ZDO security remove device extended extension message.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 8 |
| --- | --- | --- | --- |
| Length = 0x08 | Cmd0 = 0x25 | Cmd1 = 0x44 | Extended Adress |

**Attributes**:

| Attribute | Length (byte) | Description |
| --- | --- | --- |

Extended Address                    8          extended address

**SRSP:**

| Byte: 1 | 1 | 1 | 1 |
|---------|---|---|---|
| Length = 0x01 | Cmd0 = 0x65 | Cmd1 = 0x44 | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|-----------|---------------|-------------|
| Status | 1 | Status of device remove |

### 3.12.1.37  ZDO_EXT_ROUTE_DISC

**Description:**

This command handles the ZDO route discovery extension message.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 2 | 1 | 1 |
|---------|---|---|---|---|---|
| Length = 0x04 | Cmd0 = 0x25 | Cmd1 = 0x45 | Destination Address | Options | Radius |

**Attributes**:

| Attribute | Length (byte) | Description |
|-----------|---------------|-------------|
| Destination Address | 2 | Destination address |
| Options | 1 | Route options |
| Radius | 1 | Broadcast radius |

**SRSP:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x65 | Cmd1 = 0x45 | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status of route discovery |

### 3.12.1.38   ZDO_EXT_ROUTE_CHECK

**Description:**

This command handles the ZDO route check extension message.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 2 | 1 | 1 |
|---|---|---|---|---|---|
| Length = 0x04 | Cmd0 = 0x25 | Cmd1 = 0x46 | Destination Address | RTStatus | Options |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Destination Address | 2 | Destination address |
| RtStatus | 1 | Status value for routing entries |
| Options | 1 | Route options |

**SRSP:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x65 | Cmd1 = 0x46 | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status of route check |

### 3.12.1.39  ZDO_EXT_REMOVE_GROUP

**Description:**

This command handles the ZDO extended remove group extension message.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 1 | 2 |
|---|---|---|---|---|
| Length = 0x03 | Cmd0 = 0x25 | Cmd1 = 0x47 | Endpoint | GroupID |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Endpoint | 1 | Endpoint to look for |
| GroupID | 2 | ID to look for group |

**SRSP:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x65 | Cmd1 = 0x47 | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status of remove group |

### 3.12.1.40  ZDO_EXT_REMOVE_ALL_GROUP

**Description:**

This command handles the ZDO extended remove all group extension message.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x25 | Cmd1 = 0x48 | Endpoint |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Endpoint | 1 | Endpoint to look for |

**SRSP:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x65 | Cmd1 = 0x48 | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status of remove all groups |

## 3.12.1.41  ZDO_EXT_FIND_ALL_GROUPS_ENDPOINT
**Description:**

This command handles the ZDO extension find all groups for endpoint message

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 1 | 2 |
|---|---|---|---|---|
| Length = 0x03 | Cmd0 = 0x25 | Cmd1 = 0x49 | Endpoint | GroupList |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Endpoint | 1 | Endpoint to look for |
| GroupList | 2 | List to hold group IDs |

**SRSP:**

| Byte: 1 | 1 | 1 | 1+2*(number of groups) |
|---|---|---|---|
| Length = 0x01+2*(number of groups) | Cmd0 = 0x65 | Cmd1 = 0x49 | Groups |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Groups | 1+2*(number of groups) | Number of groups copied to group list |

## 3.12.1.42  ZDO_EXT_FIND_GROUP

**Description:**

This command handles the ZDO extension find group message

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 1 | 2 |
|---|---|---|---|---|
| Length = 0x03 | Cmd0 = 0x25 | Cmd1 = 0x4A | Endpoint | GroupID |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Endpoint | 1 | Endpoint to look for |
| GroupID | 2 | Group ID |

**SRSP:**

| Byte: 1 | 1 | 1 | 0-18 |
|---|---|---|---|
| Length = 0x01-0x13 | Cmd0 = 0x65 | Cmd1 = 0x4A | Group |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Group | 0-18 | Group information |

### 3.12.1.43  ZDO_EXT_ADD_GROUP
**Description:**

This command handles the ZDO extension add group message.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 1 | 2 | 16 |
|---|---|---|---|---|---|
| Length = 0x013 | Cmd0 = 0x25 | Cmd1 = 0x4B | Endpoint | GroupID | GroupName |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Endpoint | 1 | Endpoint to look for |
| GroupID | 2 | Group ID |
| GroupName | 16 | Group name |

**SRSP:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x65 | Cmd1 = 0x4B | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status of adding group |

### 3.12.1.44  ZDO_EXT_COUNT_ALL_GROUPS
**Description:**

This command handles the ZDO extension count all groups message.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 |
|---|---|---|
| Length = 0x00 | Cmd0 = 0x25 | Cmd1 = 0x4C |

**Attributes**:

| Attribute | Length (byte) | Description |
|-----------|---------------|-------------|
| None | | |

**SRSP:**

| Byte: 1 | 1 | 1 | 1 |
|---------|---|---|---|
| Length = 0x01 | Cmd0 = 0x65 | Cmd1 = 0x4C | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|-----------|---------------|-------------|
| Status | 1 | Total number of groups |

### 3.12.1.45   ZDO_EXT_RX_IDLE

**Description:**

This command handles the ZDO extension Get/Set RxOnIdle to ZMac message

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 1 | 1 |
|---------|---|---|---|---|
| Length = 0x02 | Cmd0 = 0x25 | Cmd1 = 0x4D | SetFlag | SetValue |

**Attributes**:

| Attribute | Length (byte) | Description |
|-----------|---------------|-------------|
| SetFlag | 1 | Set or get the value |
| SetValue | 1 | Value to be set to ZMac message |

**SRSP:**

| Byte: 1 | 1 | 1 | 1 |
|---------|---|---|---|
| Length = 0x01 | Cmd0 = 0x65 | Cmd1 = 0x4D | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|-----------|---------------|-------------|
| Status | 1 | Status of get or set value |

### 3.12.1.46  ZDO_EXT_UPDATE_NWK_KEY

**Description:**

This command handles the ZDO security update network key extension message.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 2 | 1 | 128 |
|---|---|---|---|---|---|
| Length = 0x083 | Cmd0 = 0x25 | Cmd1 = 0x4E | Destination Address | KeySeqNum | Key |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| DestinationAddress | 2 | Destination address |
| KeySeqNum | 1 | Key Sequence number |
| Key | 128 | Network key |

**SRSP:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x65 | Cmd1 = 0x4E | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status of key update |

### 3.12.1.47  ZDO_EXT_SWITCH_NWK_KEY

**Description:**

This command handles the ZDO security switch network key extension message.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 2 | 1 |
|---|---|---|---|---|
| Length = 0x03 | Cmd0 = 0x25 | Cmd1 = 0x4F | Destination Address | KeySeqNum |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| DestinationAddress | 2 | Destination address |
| KeySeqNum | 1 | Key Sequence number |

**SRSP:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x65 | Cmd1 = 0x4F | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status of key switch |

## 3.12.1.48  ZDO_EXT_NWK_INFO

**Description:**

This command handles the ZDO extension network message.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 |
|---|---|---|
| Length = 0x00 | Cmd0 = 0x25 | Cmd1 = 0x50 |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| None | | |

**SRSP:**

| Byte: 1 | 1 | 1 | 2 | 2 | 2 | 8 | 8 | 2 |
|---|---|---|---|---|---|---|---|---|
| Length = 0x18 | Cmd0 = 0x65 | Cmd1 = 0x50 | Short Address | PAN ID | Parent address | Extended PAN ID | Extended Parent Address | Channel |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|

Copyright © 2008-2016 Texas Instruments, Inc.  All rights reserved.

| Short Address | 2 | Short address |
|---|---|---|
| PAN ID | 2 | PAN ID |
| Parent Address | 2 | Parent address |
| Extended PAN ID | 8 | Extended PAN ID |
| Extended Parent Address | 8 | IEEE address of parent |
| Channel | 2 | Current channel |

## 3.12.1.49   ZDO_EXT_SEC_APS_REMOVE_REQ

**Description:**

This command handles the ZDO extension Security Manager APS Remove Request message.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 2 | 8 | 2 |
|---|---|---|---|---|---|
| Length = 0x0c | Cmd0 = 0x25 | Cmd1 = 0x51 | NWK Address | Extended Address | Parent address |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| NWK Address | 2 | Device's network address |
| Extended Address | 8 | IEEE address of device |
| Parent Address | 2 | Parent's network address |

**SRSP:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x65 | Cmd1 = 0x51 | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|

| Status | 1 | Status of removing device from the network |
|--------|---|--------------------------------------------|

### 3.12.1.50  ZDO_FORCE_CONCENTRATOR_CHANGE

**Description:**

This command forces a network concentrator change by resetting    zgConcentratorEnable and zgConcentratorDiscoveryTime from NV and set nwk event.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 |
|---------|---|---|
| Length = 0x00 | Cmd0 = 0x25 | Cmd1 = 0x52 |

**Attributes**:

| Attribute | Length (byte) | Description |
|-----------|---------------|-------------|
| None | | |

**SRSP:**

| Byte: 1 | 1 | 1 |
|---------|---|---|
| Length = 0x00 | Cmd0 = 0x65 | Cmd1 = 0x52 |

**Attributes**:

| Attribute | Length (byte) | Description |
|-----------|---------------|-------------|
| None | | |

### 3.12.1.51  ZDO_EXT_SET_PARAMS

**Description:**

This command set parameters not settable through NV.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 1 |
|---------|---|---|---|
| Length = 0x01 | Cmd0 = 0x25 | Cmd1 = 0x53 | useMulticast |

**Attributes**:

| Attribute | Length (byte) | Description |
|-----------|---------------|-------------|

| useMulticast | 1 | Set or reset the use of multicast |
|---|---|---|

**SRSP:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x65 | Cmd1 = 0x53 | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status of use of multicast |

## 3.12.1.52  ZDO_NWK_ADDR_OF_INTEREST_REQ

**Description:**

This command handles ZDO network address of interest request.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 2 | 2 | 1 |
|---|---|---|---|---|---|
| Length = 0x05 | Cmd0 = 0x25 | Cmd1 = 0x29 | destAddr | nwkAddrofIntere st | cmd |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| destAddr | 2 | Destination Address |
| nwkAddrofInterest | 2 | 16 bit network address of interest |
| cmd | 1 | A valid cluster ID as specified by the Profile |

**SRSP:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x65 | Cmd1 = 0x29 | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|

| Status | 1 | | | AF status | |

## 3.12.2   MT_ZDO Callbacks

### 3.12.2.1   ZDO_NWK_ADDR_RSP

**Description:**

This command is issued by the tester to return the results from a ZDO_NWK_ADDR_REQ.

**Usage**

**AREQ:**

| 1 | 1 | 1 | 1 | 8 | 2 |
|---|---|---|---|---|---|
| Length = 0x0D-0x53 | Cmd0 = 0x45 | Cmd1 = 0x80 | Status | IEEEAddr | NwkAddr |

| 1 | 1 | 0-70 |
|---|---|---|
| StartIndex | NumAssocDev | AssocDevList |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | This field indicates either SUCCESS or FAILURE. |
| IEEEAddr | 8 | 64 bit IEEE address of source device. |
| NwkAddr | 2 | Specifies the short network address of responding device. |
| StartIndex | 1 | Specifies the starting index into the list of associated devices for this report. |
| NumAssocDev | 1 | Specifies the number of associated devices. |
| AssocDevList | 0-70 | Contains the list of network address for associated devices. This list can be a partial list if the entire list doesn't fit into a packet. If it is a partial list, the starting index is StartIndex. |

### 3.12.2.2    ZDO_IEEE_ADDR_RSP

**Description:**

This callback message is in response to the ZDO IEEE Address Request.


Usage:

**AREQ:**

| 1 | 1 | 1 | 1 | 8 | 2 |
|---|---|---|---|---|---|
| Length = 0x0D-0x53 | Cmd0 = 0x45 | Cmd1 = 0x81 | Status | IEEEAddr | NwkAddr |

| 1 | 1 | 0-70 |
|---|---|---|
| StartIndex | NumAssocDev | AssocDevList |


**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | This field indicates either SUCCESS or FAILURE. |
| IEEEAddr | 8 | 64 bit IEEE address of source device. |
| NwkAddr | 2 | Specifies the short network address of responding device. |
| StartIndex | 1 | Specifies the starting index into the list of associated devices for this report. |
| NumAssocDev | 1 | Specifies the number of associated devices. |
| AssocDevList | 0-70 | Contains the list of network address for associated devices.  This list can be a partial list if the entire list doesn't fit into a packet.  If it is a partial list, the starting index is StartIndex. |

### 3.12.2.3    ZDO_NODE_DESC_RSP

**Description:**

This callback message is in response to the ZDO Node Descriptor Request.

**Usage:**

**AREQ:**

| 1 | 1 | 1 | 2 | 1 | 2 |
|---|---|---|---|---|---|
| Length = 0x12 | Cmd0 = 0x45 | Cmd1 = 0x82 | SrcAddr | Status | NwkAddr |

| 1 | 1 | 1 | 2 |
|---|---|---|---|
| LogicalType/ ComplexDescAvailable/ UserDescAvailable | APSFlags/ FrequencyBand | MACCapabilityFlags | ManufacturerCode |

| 1 | 2 | 2 | 2 | 1 |
|---|---|---|---|---|
| MaxBufferSize | MaxTransferSize | ServerMask | MaxOutTransferSize | DescriptorCapabilities |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| SrcAddr | 2 | The message's source network address. |
| Status | 1 | This field indicates either SUCCESS or FAILURE. |
| NWKAddrOfInterest | 2 | Device's short address of this Node descriptor |
| LogicalType/ ComplexDescriptorAvailable/ UserDescriptorAvailable | 1 | Logical Type: Bit 0-2 –<br><br>| Description | Value |<br>|---|---|<br>| ZigBee Coordinator | 0 |<br>| ZigBee Router | 1 |<br>| ZigBee End Device | 2 |<br><br>ComplexDescriptorAvailable: Bit 3 – Indicates if complex descriptor is available for the node<br><br>UserDescriptorAvailable: Bit 4 – Indicates if user descriptor is available for the node<br><br>Reserved: Bit 5-7 – Reserved for future use |
| APSFlags/FrequencyBand | 1 | APSFlags – Bit 0-4 – Node Flags assigned for APS. For V1.0 all bits are reserved.<br><br>NodeFrequencyBand – Bit 5-7 – Identifies node frequency band capabilities |

Capability flags stored for the MAC:

| Description | Value |
|---|---|
| CAPINFO_DEVICETYPE_RFD | 0x00 |
| CAPINFO_ALTPANCOORD | 0x01 |
| CAPINFO_DEVICETYPE_FFD | 0x02 |
| CAPINFO_POWER_AC | 0x04 |
| CAPINFO_RCVR_ON_IDLE | 0x08 |
| CAPINFO_SECURITY_CAPABLE | 0x40 |
| CAPINFO_ALLOC_ADDR | 0x80 |

MacCapabilitiesFlags — 1

ManufacturerCode — 2 — Specifies a manufacturer code that is allocated by the ZigBee Alliance, relating to the manufacturer to the device.

MaxBufferSize — 1 — Indicates size of maximum NPDU. This field is used as a high level indication for management.

MaxInTransferSize — 2 — Indicates maximum size of Transfer up to 0x7fff (This field is reserved in version 1.0 and shall be set to zero).

ServerMask — 2

Bit 0 - Primary Trust Center

  1 - Backup Trust Center

  2 - Primary Binding Table Cache

  3 - Backup Binding Table Cache

  4 - Primary Discovery Cache

  5 - Backup Discovery Cache

MaxOutTransferSize — 2 — Indicates maximum size of Transfer up to 0x7fff (This field is reserved in version 1.0 and shall be set to zero).

DescriptorCapabilities — 1 — Specifies the Descriptor capabilities

### 3.12.2.4    ZDO_POWER_DESC_RSP

**Description:**

This callback message is in response to the ZDO Power Descriptor Request.

**Usage:**

**AREQ:**

| 1 | 1 | 1 | 2 | 1 | 2 |
|---|---|---|---|---|---|
| Length = 0x07 | Cmd0 = 0x45 | Cmd1 = 0x83 | SrcAddr | Status | NwkAddr |

| 1 | 1 |
|---|---|
| CurrentPowerMode/AvailablePowerSources | CurrentPowerSource/CurrentPowerSourceLevel |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| SrcAddr | 2 | Specifies the message's source network address. |
| Status | 1 | This field indicates either SUCCESS or FAILURE. |
| NWKAddr | 2 | Specifies Device's short address that this response describes. |
| CurrentPowerMode/AvailablePowerSources | 1 | - CurrentPowerMode: bits 3-0<br>- AvailablePowerSources: bits 7-4 |
| CurrentPowerSource/CurrentPowerSourceLevel | 1 | - CurrentPowerSource: bits 3-0<br>- CurrentPowerSourceLevel: bits 7-4 |

### 3.12.2.5   ZDO_SIMPLE_DESC_RSP

**Description:**

This callback message is in response to the ZDO Simple Descriptor Request

**Usage:**

**AREQ:**

| 1 | 1 | 1 | 2 | 1 | 2 | 1 |
|---|---|---|---|---|---|---|
| Length = 0x06-4E | Cmd0 = 0x45 | Cmd1 = 0x84 | SrcAddr | Status | NwkAddr | Len |

| 1 | 2 | 2 | 1 |
|---|---|---|---|
| Endpoint | ProfileId | DeviceId | DeviceVersion |

| 1 | 0-32 | 1 | 0-32 |
|---|---|---|---|
| NumInClusters | InClusterList | NumOutClusters | OutClusterList |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| SrcAddr | 2 | Specifies the message's source network address. |
| Status | 1 | This field indicates either SUCCESS or FAILURE. |
| NWKAddr | 2 | Specifies Device's short address that this response describes. |
| Len | 1 | Specifies the length of the simple descriptor |
| Endpoint | 1 | Specifies Endpoint of the device |
| ProfileId | 2 | The profile Id for this endpoint. |
| DeviceId | 2 | The Device Description Id for this endpoint. |
| DeviceVersion | 1 | Defined as the following format<br>0 – Version 1.00<br>0x01-0x0F – Reserved. |
| NumInClusters | 1 | The number of input clusters in the InClusterList. |
| InClusterList | 0-32 | List of input cluster Id's supported. |
| NumOutClusters | 1 | The number of output clusters in the OutClusterList. |
| OutClusterList | 0-32 | List of output cluster Id's supported. |

### 3.12.2.6    ZDO_ACTIVE_EP_RSP

**Description:**

This callback message is in response to the ZDO Active Endpoint Request.

**Usage:**

**AREQ:**

| 1 | 1 | 1 | 2 | 1 | 2 | 1 |
|---|---|---|---|---|---|---|
| Length = 0x06-0x53 | Cmd0 = 0x45 | Cmd1 = 0x85 | SrcAddr | Status | NwkAddr | ActiveEPCount |

| 0-77 |
|------|
| ActiveEPList |

**Attributes:**

| Attribute | Length (byte) | Description |
|-----------|---------------|-------------|
| SrcAddr | 2 | The message's source network address. |
| Status | 1 | This field indicates either SUCCESS or FAILURE. |
| NWKAddr | 2 | Device's short address that this response describes. |
| ActiveEPCount | 1 | Number of active endpoint in the list |
| ActiveEPList | 0-77 | Array of active endpoints on this device. |

### 3.12.2.7    ZDO_MATCH_DESC_RSP

**Description:**

This callback message is in response to the ZDO Match Descriptor Request.

Usage:

**AREQ:**

| 1 | 1 | 1 | 2 | 1 | 2 | 1 |
|---|---|---|---|---|---|---|
| Length = 0x06-0x53 | Cmd0 = 0x45 | Cmd1 = 0x86 | SrcAddr | Status | NwkAddr | MatchLength |

| 0-77 |
|---|
| MatchList |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| SrcAddr | 2 | The message's source network address. |
| Status | 1 | This field indicates either SUCCESS or FAILURE. |
| NWKAddr | 2 | Device's short address that this response describes. |
| MatchLength | 1 | The count of endpoints on the remote device that match the request criteria |
| MatchList | 0-77 | List of bytes, each represents an 8 bit endpoint |

### 3.12.2.8    ZDO_COMPLEX_DESC_RSP

**Description:**

This callback message is in response to the ZDO Complex Descriptor Request.

**Usage:**

**AREQ:**

| 1 | 1 | 1 | 2 | 1 | 2 | 1 |
|---|---|---|---|---|---|---|
| Length = 0x06-0x53 | Cmd0 = 0x45 | Cmd1 = 0x87 | SrcAddr | Status | NwkAddr | ComplexLength |

| 0-77 |
|---|
| ComplexList |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| SrcAddr | 2 | The message's source network address. |
| Status | 1 | This field indicates either SUCCESS or FAILURE. |
| NWKAddr | 2 | Device's short address that this response describes. |
| ComplexLength | 1 | Length of the complex descriptor. |
| ComplexDescriptor | 0-77 | Array of bytes contains the complex descriptor. |

### 3.12.2.9    ZDO_USER_DESC_RSP

**Description:**

This callback message is in response to the ZDO User Descriptor Request.

**Usage:**

**AREQ:**

| 1 | 1 | 1 | 2 | 1 | 2 | 1 | 0-77 |
|---|---|---|---|---|---|---|---|
| Length = 0x06-0x16 | Cmd0 = 0x45 | Cmd1 = 0x88 | SrcAddr | Status | NwkAddr | Len | UserDescriptor |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| SrcAddr | 2 | The message's source network address. |
| Status | 1 | This field indicates either SUCCESS or FAILURE. |
| NWKAddr | 2 | Device's short address that this response describes. |
| UserLength | 1 | Length of the complex descriptor. |
| UserDescriptor | 0-77 | Array of bytes contains user descriptor. |

### 3.12.2.10   ZDO_USER_DESC_CONF

**Description:**

This confirmation notifies the user when the device receives a user descriptor.

**Usage:**

**AREQ:**

| 1 | 1 | 1 | 2 | 1 | 2 |
|---|---|---|---|---|---|
| Length = 0x05 | Cmd0 = 0x45 | Cmd1 = 0x89 | SrcAddr | Status | NwkAddr |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| SrcAddr | 2 | The message's source network address. |
| Status | 1 | This field indicates either SUCCESS or FAILURE. |
| NWKAddr | 2 | Device's short address that this response describes. |

## 3.12.2.11  ZDO_SERVER_DISC_RSP

**Description:**

This callback message is in response to the ZDO System Service Discovery Request. Upon receiving the request, 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.

**Usage:**

**AREQ:**

| 1 | 1 | 1 | 2 | 1 | 2 |
|---|---|---|---|---|---|
| Length = 0x05 | Cmd0 = 0x45 | Cmd1 = 0x8A | SrcAddr | Status | ServerMask |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| SrcAddr | 2 | The message's source network address. |
| Status | 1 | This field indicates either SUCCESS (0) or FAILURE (1). |
| Server Mask | 2 | Each bit signifies one system server capability of the node. The bit setting is defined in the following table: |

| Bit Number | Assignment |
|---|---|
| 0 | Primary Trust Center |
| 1 | Backup Trust Center |
| 2 | Primary Binding Table Cache |
| 3 | Backup Binding Table Cache |
| 4 | Primary Discovery Cache |
| 5 | Backup Discovery Cache |
| 6– 15 | Reserved |

### 3.12.2.12  ZDO_END_DEVICE_BIND_RSP

**Description:**

This callback message is in response to the ZDO End Device Bind Request.

Usage:

**AREQ:**

| 1 | 1 | 1 | 2 | 1 |
|---|---|---|---|---|
| Length = 0x03 | Cmd0 = 0x45 | Cmd1 = 0xA0 | SrcAddr | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| SrcAddr | 2 | The message's source network address. |
| Status | 1 | This field indicates either SUCCESS (0) or FAILURE (1). |

### 3.12.2.13  ZDO_BIND_RSP

**Description:**

This callback message is in response to the ZDO Bind Request.

**Usage:**

**AREQ:**

| 1 | 1 | 1 | 2 | 1 |
|---|---|---|---|---|
| Length = 0x03 | Cmd0 = 0x45 | Cmd1 = 0xA1 | SrcAddr | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| SrcAddr | 2 | The message's source network address. |
| Status | 1 | This field indicates either SUCCESS (0) or FAILURE (1). |

### 3.12.2.14  ZDO_UNBIND_RSP

**Description:**

This callback message is in response to the ZDO Unbind Request.

**Usage:**

**AREQ:**

| Byte: 1 | 1 | 1 | 2 | 1 |
|---|---|---|---|---|
| Length = 0x03 | Cmd0 = 0x45 | Cmd1 = 0xA2 | SrcAddr | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| SrcAddr | 2 | The message's source network address. |
| Status | 1 | This field indicates either SUCCESS (0) or FAILURE (1). |

### 3.12.2.15  ZDO_MGMT_NWK_DISC_RSP

**Description:**

This callback message is in response to the ZDO Management Network Discovery Request.

**Usage:**

**AREQ:**

| 1 | 1 | 1 | 2 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|
| Length = 0x06-0x4E | Cmd0 = 0x45 | Cmd1 = 0xB0 | SrcAddr | Status | NetworkCount | StartIndex |

| 1 | 0-72 |
|---|---|
| NetworkListCount | NetworkList Records |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| SrcAddr | 2 | Source address of the message. |
| Status | 1 | This field indicates either SUCCESS or FAILURE. |
| NetworkCount | 1 | Total number of entries available in the device. |
| StartIndex | 1 | Where in the total number of entries this response starts. |
| NetworkListCount | 1 | Number of entries in this response. |

An array of NetworkList items. NetworkListCount contains the number of items in this table

| | | Name | Size | Description |
|---|---|---|---|---|
| NetworkList | List | PAN ID/Extended PAN ID | 2 bytes | PAN ID of the neighbor device |
| | | Logical Channel | 1 byte | The current logical channel occupied by the network. |
| | | Stack Profile / ZigBee Version | 1 byte | StackProfile: bits 3-0 <br><br> ZigBeeVersion: bits 7-4 <br><br> A ZigBee stack profile Identifier indicating stack profile in use in the discovered network <br><br> The version of the ZigBee protocol in use in the discovered network. |
| | | Beacon Order / Super frame Order | 1 byte | BeaconOrder: bits 3-0 <br><br> SuperframeOrder: bits 7-4 |
| | | Permit Joining | 1 byte | Permit joining flag |

## 3.12.2.16   ZDO_MGMT_LQI_RSP

**Description:**

This callback message is in response to the ZDO Management LQI Request.

**Usage:**

**AREQ:**
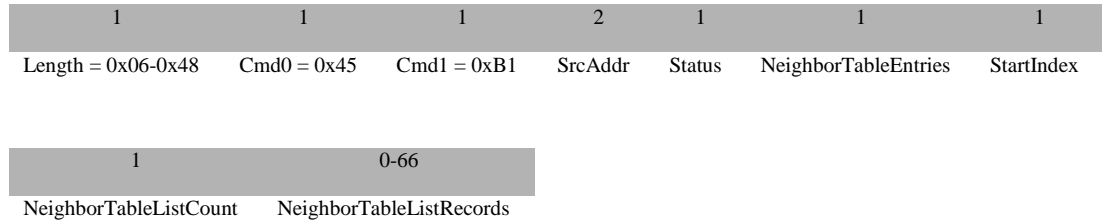
| 1 | 1 | 1 | 2 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|
| Length = 0x06-0x48 | Cmd0 = 0x45 | Cmd1 = 0xB1 | SrcAddr | Status | NeighborTableEntries | StartIndex |

| 1 | 0-66 |
|---|---|
| NeighborTableListCount | NeighborTableListRecords |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| SrcAddr | 2 | Source address of the message. |
| Status | 1 | This field indicates either SUCCESS or FAILURE. |
| NeighborTableEntries | 1 | Total number of entries available in the device. |
| StartIndex | 1 | Where in the total number of entries this response starts. |

NeighborLqiListCount          1          Number of entries in this response.

An array of NeighborLqiList items. NeighborLQICount contains the number of items in this table.

| Name | Size | Description |
|------|------|-------------|
| ExtendedPanID | 8 bytes | Extended PAN ID of the neighbor device |
| ExtendedAddress | 8 bytes | Network extended address |
| NetworkAddress | 2 bytes | Device short address |
| DeviceType/ RxOnWhenIdle/ Relationship | 1 byte | DeviceType: bits 1-0  RxOnWhenIdle: bits 3-2  Relationship: bits 6-4 |
| PermitJoining | 1 byte | PermitJoining: bits 1-0 |
| Depth | 1 byte | |
| LQI | 1 byte | |

NeighborLqiList          0-66

### 3.12.2.17  ZDO_MGMT_RTG_RSP

**Description:**

This callback message is in response to the ZDO Management Routing Table Request.

**Usage:**

**AREQ:**

| 1 | 1 | 1 | 2 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|
| Length = 0x06-0x51 | Cmd0 = 0x45 | Cmd1 = 0xB2 | SrcAddr | Status | RoutingTableEntries | StartIndex |

| 1 | 0-75 |
|---|------|
| RoutingTableListCount | RoutingTableListRecords |

**Attributes:**

| Attribute | Length (byte) | Description |
|-----------|---------------|-------------|
| SrcAddr | 2 | Source address of the message. |
| Status | 1 | This field indicates either SUCCESS or FAILURE. |
| RoutingTableEntries | 1 | Total number of entries available in the device. |

StartIndex             1        Where in the total number of entries this response starts.

RoutingTableListCount  1        Number of entries in this response.

                                An array of RtgList items. RtgListCount contains the number
                                of items in this table.

| Name | Size | Description |
|---|---|---|
| Destination Address | 2 bytes | Network destination address |
| Status | 1 byte | Route status: bits 2-0<br><br>0x00 Active<br>0x01 Discovery Underway<br>0x02 Discovery Failed<br>0x03 Inactive<br>0x04 – 0x07 Reserved |
| Next Hop | 2 bytes | Next hop network address |

RoutingTableList       0-75

## 3.12.2.18  ZDO_MGMT_BIND_RSP

**Description:**

This callback message is in response to the ZDO Management Binding Table Request.

**Usage:**

**AREQ:**

| Byte: 1 | 1 | 1 | 2 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|
| Length = 0x06-0x51 | Cmd0 = 0x45 | Cmd1 = 0xB3 | SrcAddr | Status | BindingTableEntries | StartIndex |

| 1 | 0-75 |
|---|---|
| BindingTableListCount | BindingTableListRecords |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| SrcAddr | 2 | Source address of the message |
| Status | 1 | This field indicates either SUCCESS (0) or FAILURE (1). |
| BindTableEntries | 1 | Total number of entries available in the device. |
| StartIndex | 1 | Where in the total number of entries this response starts. |

| | | | |
|---|---|---|---|
| BindTableListCount | 1 | Number of entries in this response. | |

An array of BindList items. BindListCount contains the number of items in this table.

| Name | Size | Description |
|---|---|---|
| SrcAddr | 8 bytes | Binding Entry's source IEEE address |
| SrcEndpoint | 1 byte | Binding Entry's source endpoint |
| ClusterId | 1 byte | Message Id in binding table |
| DstAddrMode | 1 byte | Address mode for binding entry's destination address |
| DstAddr | 8 bytes | Binding Entry's destination IEEE address |
| DstEndpoint | 1 byte | Binding Entry's destination endpoint. For V1.1, this field is only present when the DestAddrMode is 64-bits extended address. |

(BindTableList — List)

### 3.12.2.19  ZDO_MGMT_LEAVE_RSP

**Description:**

This callback message is in response to the ZDO Management Leave Request.

**Usage:**

**AREQ:**

| Byte: 1 | 1 | 1 | 2 | 1 |
|---|---|---|---|---|
| Length = 0x03 | Cmd0 = 0x45 | Cmd1 = 0xB4 | SrcAddr | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| SrcAddr | 2 | Source address of the message |
| Status | 1 | This field indicates either SUCCESS (0) or FAILURE (1). |

### 3.12.2.20  ZDO_MGMT_DIRECT_JOIN_RSP

**Description:**

This callback message is in response to the ZDO Management Direct Join Request.

**Usage:**

**AREQ:**

| 1 | 1 | 1 | 2 | 1 |
|---|---|---|---|---|

| Length = 0x03 | Cmd0 = 0x45 | Cmd1 = 0xB5 | SrcAddr | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|-----------|---------------|-------------|
| SrcAddr | 2 | Source address of the message |
| Status | 1 | This field indicates either SUCCESS (0) or FAILURE (1). |

### 3.12.2.21  ZDO_MGMT_PERMIT_JOIN_RSP

**Description:**

This callback message is in response to the ZDO Management Permit Join Request.

**Usage:**

**AREQ:**

| 1 | 1 | 1 | 2 | 1 |
|---|---|---|---|---|
| Length = 0x03 | Cmd0 = 0x45 | Cmd1 = 0xB6 | SrcAddr | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|-----------|---------------|-------------|
| SrcAddr | 2 | Source address of the message. |
| Status | 1 | This field indicates either SUCCESS (0) or FAILURE (1). |

### 3.12.2.22  ZDO_STATE_CHANGE_IND

**Description:**

This callback message indicates the ZDO state change.

**Usage:**

**AREQ:**

| 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x45 | Cmd1 = 0xC0 | State |

**Attributes:**

| Attribute | Length (byte) | Description |
|-----------|---------------|-------------|
| State | 1 | Specifies the changed ZDO state. |

191        Copyright © 2008-2016 Texas Instruments, Inc.  All rights reserved.

### 3.12.2.23   ZDO_END_DEVICE_ANNCE_IND

**Description:**

This callback indicates the ZDO End Device Announce.


Usage:

**AREQ:**

| Byte: 1 | 1 | 1 | 2 | 2 | 8 | 1 |
|---|---|---|---|---|---|---|
| Length = 0x0D | Cmd0 = 0x45 | Cmd1 = 0xC1 | SrcAddr | NwkAddr | IEEEAddr | Capabilites |


**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| SrcAddr | 2 | Source address of the message. |
| NwkAddr | 2 | Specifies the device's short address. |
| IEEEAddr | 8 | Specifies the 64 bit IEEE address of source device. |
| Capabilites | 1 | Specifies the MAC capabilities of the device.<br><br>Bit: 0 – Alternate PAN Coordinator<br><br>1 – Device type: 1- ZigBee Router; 0 – End Device<br><br>2 – Power Source: 1 Main powered<br><br>3 – Receiver on when Idle<br><br>4 – Reserved<br><br>5 – Reserved<br><br>6 – Security capability<br><br>7 – Reserved |

### 3.12.2.24  ZDO_MATCH_DESC_RSP_SENT

**Description:**

This callback indicates that Match Descriptor Response has been sent.

**Usage:**

**AREQ:**

| Byte: 1 | 1 | 1 | 2 |
|---|---|---|---|
| Length = 0x04-0x44 | Cmd0 = 0x45 | Cmd1 = 0xC2 | NwkAddr |

| 1 | 0-32 | 1 | 0-32 |
|---|---|---|---|
| NumInClusters | InClusterList | NumOutClusters | OutClusterList |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| NwkAddr | 2 | Specifies the device's short address |
| NumInClusters | 1 | The number of input clusters in the InClusterList. |
| InClusterList | 0-32 | List of input cluster Id's supported. |
| NumOutClusters | 1 | The number of output clusters in the OutClusterList. |
| OutClusterList | 0-32 | List of output cluster Id's supported. |

### 3.12.2.25  ZDO_STATUS_ERROR_RSP

**Description:**

This message is the default message for error status.

**Usage:**

**AREQ:**

| Byte: 1 | 1 | 1 | 2 | 1 |
|---|---|---|---|---|
| Length = 0x04-0x44 | Cmd0 = 0x45 | Cmd1 = 0xC3 | SrcAddr | Status |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| SrcAddr | 2 | Source address of the message |
| Status | 1 | This field indicates either SUCCESS (0) or FAILURE (1). |

### 3.12.2.26  ZDO_SRC_RTG_IND

**Description:**

This message is an indication to inform host device the receipt of a source route to a given device.

**Usage:**

**AREQ:**

| Byte: 1 | 1 | 1 | 2 | 1 | 2N |
|---|---|---|---|---|---|
| Length = 0x04-0x44 | Cmd0 = 0x45 | Cmd1 = 0xC4 | dstAddr | Relay Count (N) | Relay List |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| DstAddr | 2 | Short address of the destination of the source route |
| Relay Count | 1 | This field indicates number of devices in the relay list of the source route. |
| Relay List | 2N | This field contains the list of devices in the relay list of the source route. It includes a two bytes short address for each device. |

### 3.12.2.27  ZDO_BEACON_NOTIFY_IND

**Description:**

This message is an indication to inform host device the receipt of a beacon notification.

**Usage:**

**AREQ:**

| Byte: 1 | 1 | 1 | 1 | N*21 |
|---|---|---|---|---|
| Length = 21 | Cmd0 = 0x45 | Cmd1 = 0xC5 | BeaconCount | BeaconList |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| BeaconCount | 1 | Number of beacons in the packet. |

An array of BeaconList items. BeaconCount contains the number of items in this table.

BeaconList

N*21

(N is BeaconCount)

| Name | Size | Description |
|------|------|-------------|
| Source Address | 2 | Short address of the source device of the beacon |
| Pan ID | 2 | ID of the  PAN |
| Logical Channel | 1 | Channel where the PAN is located. |
| Permit Joining | 1 | Flag to indicate whether the device accept association. |
| Router Capacity | 1 | Flag to indicate whether the device accept other router to associate |
| Device Capacity | 1 | Flag to indicate whether the device accept other device to associate |
| Protocol Version | 1 | Version of the ZigBee protocol. Value '1' represents ZigBee 2004. Value '2' represents ZigBee 2006/2007 |
| Stack Profile | 1 | Stack profile of the PAN: Stack Profile 1 for ZigBee and Stack Profile 2 for ZigBee Pro. |
| LQI | 1 | LQI (Link quality indicator) measurement of the beacon. |
| Depth | 1 | Depth of the source device, i.e. number of hops from the device to the ZigBee coordinator. |
| Update ID | 1 | Update ID of the device. |
| Extended Pan ID | 8 | 64 bit extended Pan ID of the Pan. |

### 3.12.2.28  ZDO_JOIN_CNF

**Description:**

This message is an indication to inform host device the result of a ZDO join request.

**Usage:**

**AREQ:**

| Byte: 1 | 1 | 1 | 1 | 2 | 2 |
|---|---|---|---|---|---|
| Length = 0x05 | Cmd0 = 0x45 | Cmd1 = 0xC6 | status | Device Address | Parent Address |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Return status of the join request: Success (0) ZMAC_NO_ACK (0xE9) if the chosen parent device dId not respond to the association request. |
| Device Address | 2 | Short address of the device. |
| Parent Address | 2 | Short address of the parent device |

### 3.12.2.29  ZDO_NWK_DISCOVERY_CNF

**Description:**

This message is an indication to inform host device the completion of network discovery scan.

**Usage:**

**AREQ:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x45 | Cmd1 = 0xC7 | status |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Return status of the network discovery. Success (0) ZMAC_NO_BEACON (0xEA) ZMAC_INVALID_PARAMETER (0xE8) if input parameter is out of valid range. |

### 3.12.2.30  ZDO_LEAVE_IND

**Description:**

This message is an indication to inform the host of a device leaving the network.

**Usage:**

**AREQ:**

| Byte: 1 | 1 | 1 | 2 | 8 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|
| Length=0x0D | Cmd0=0x45 | Cmd1=0xC9 | SrcAddr | ExtAddr | Request | Remove | Rejoin |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| SrcAddr | 2 | Short address (LSB-MSB) of the source of the leave indication. |
| ExtAddr | 8 | Extended address (LSB-MSB) of the source of the leave indication. |
| Request | 1 | Boolean, TRUE = request, FALSE = indication. |
| Remove | 1 | Boolean, TRUE = remove children. |
| Rejoin | 1 | Boolean, TRUE = rejoin. |

### 3.12.2.31  ZDO_MSG_CB_INCOMING

**Description:**

This message is a ZDO callback for a Cluster Id that the host requested to receive with a ZDO_ MSG_CB_REGISTER request.

**Usage:**

**AREQ:**

| Byte: 1 | 1 | 1 | 2 | 1 | 2 |
|---|---|---|---|---|---|
| Length = 0x09-Max MTU length | Cmd0 = 0x45 | Cmd1 = 0xFF | SrcAddr | WasBroadcast | ClusterId |

| 1 | 1 | 2 | 0-Max MTU length |
|---|---|---|---|
| SecurityUse | SeqNum | MacDstAddr | Data |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| SrcAddr | 2 | Short address (LSB-MSB) of the source of the ZDO message. |
| WasBroadcast | 1 | This field indicates whether or not this ZDO message was broadcast. |

| ClusterId | 2 | The ZDO Cluster Id of this message. |
|---|---|---|
| SecurityUse | 1 | N/A – not used. |
| SeqNum | 1 | The sequence number of this ZDO message. |
| MacDstAddr | 2 | The MAC destination short address (LSB-MSB) of the ZDO message. |
| Data | 0-Max MTU length. | The data that corresponds to the Cluster Id of the message (see reference [4], "ZDO Parsing Functions" for information on parsing the data that corresponds to each ZDO Cluster Id). |

### 3.12.2.32   ZDO_TC_DEV_IND

**Description:**

This message is a ZDO callback for TC Device Indication.

Usage:

**AREQ:**

| Byte: 1 | 1 | 1 | 2 | 8 | 2 |
|---|---|---|---|---|---|
| Length = 0x0C | Cmd0 = 0x45 | Cmd1 = 0xCA | SrcNwkAddr | SrcIEEEAddr | Parent Nwk Addr |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| SrcNwkAddr | 2 | Source network Address. |
| WasBroadcast | 8 | IEEE Address of the source |
| ParentNwkAddr | 2 | Network address of the parent |

### 3.12.2.33  ZDO_PERMIT_JOIN_IND

**Description:**

This message is a ZDO callback for Permit Join Indication.

**Usage:**

**AREQ:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x45 | Cmd1 = 0xCB | PermitJoinDuration |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Permit Join Duration | 1 | Duration for which joining is permitted |

## 3.13    MT_APP_CONFIG

This interface allows the tester to issue commands to configure some parameters of the device, trust center and BDB subsystem.

### 3.13.1  MT_APP_CONFIG Commands

#### 3.13.1.1  APP_CNF_SET_NWK_FRAME_COUNTER

**Description:**

This message sets the network frame counter to the value specified in the Frame Counter Value. For projects with multiple instances of frame counter, the message sets the frame counter of the current network.

**Usage:**

This message should only be use for debugging purposes. Changing the network frame counter in a production device, may cause the failures in the authentication process with other devices in the network.

**SREQ:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x2F | Cmd1 = 0xFF | Frame Counter Value |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Frame Counter Value | 4 | Set the network frame counter to the specified value |

**SRSP:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x6F | Cmd1 = 0xFF | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

## 3.13.1.2    APP_CNF_SET_DEFAULT_REMOTE_ENDDEVICE _TIMEOUT

**Description:**

Sets the default value used by parent device to expire legacy child devices.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x2F | Cmd1 = 0x01 | Timeout Index |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Timeout Index | 1 | Specifies the number of seconds until the timeout expires, according to the next list:<br><br>0x00    10 seconds<br><br>0x01    2 minutes<br><br>0x02    4 minutes<br><br>0x03    8 minutes<br><br>0x04    16 minutes<br><br>0x05    32 minutes |

| 0x06 | 64 minutes |
| 0x07 | 128 minutes |
| 0x08 | 256 minutes (Default) |
| 0x09 | 512 minutes |
| 0x0A | 1024 minutes |
| 0x0B | 2048 minutes |
| 0x0C | 4096 minutes |
| 0x0D | 8192 minutes |
| 0x0E | 16384 minutes |

**SRSP:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x6F | Cmd1 = 0x01 | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Invalid Parameter (2). |

## 3.13.1.3    APP_CNF_SET_ENDDEVICETIMEOUT

**Description:**

Sets in ZED the timeout value to be send to parent device for child expiring.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x2F | Cmd1 = 0x02 | Timeout Index |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Timeout Index | 1 | Specifies the number of seconds until the timeout expires, according to the next list:<br><br>0x00    10 seconds<br><br>0x01    2 minutes |

| 0x02 | 4 minutes |
| 0x03 | 8 minutes |
| 0x04 | 16 minutes |
| 0x05 | 32 minutes |
| 0x06 | 64 minutes |
| 0x07 | 128 minutes |
| 0x08 | 256 minutes (Default) |
| 0x09 | 512 minutes |
| 0x0A | 1024 minutes |
| 0x0B | 2048 minutes |
| 0x0C | 4096 minutes |
| 0x0D | 8192 minutes |
| 0x0E | 16384 minutes |

**SRSP:**

| Byte: 1 | 1 | 1 | 1 |
|---------|---|---|---|
| Length = 0x01 | Cmd0 = 0x6F | Cmd1 = 0x02 | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|-----------|---------------|-------------|
| Status | 1 | Status is either Success (0) or Invalid Parameter (2). |

### 3.13.1.4   APP_CNF_SET_ALLOWREJOIN_TC_POLICY

**Description:**

Sets the AllowRejoin TC policy.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 1 |
|---------|---|---|---|
| Length = 0x01 | Cmd0 = 0x2F | Cmd1 = 0x03 | Allow Rejoin |

**Attributes:**

| Attribute | Length (byte) | Description |
|-----------|---------------|-------------|

| Allow Rejoin | 1 | This value specifies whether or not the Trust Center allows devices to rejoin. |

**SRSP:**

| Byte: 1 | 1 | 1 | 1 |
|---------|---|---|---|
| Length = 0x01 | Cmd0 = 0x6F | Cmd1 = 0x05 | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|-----------|---------------|-------------|
| Status | 1 | Status is either Success (0) or Failure (1). |

## 3.13.1.5    APP_CNF_BDB_START_COMMISSIONING

**Description:**

Set the commissioning methods to be executed. Initialization of BDB is executed with this call, regardless of its parameters.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 1 |
|---------|---|---|---|
| Length = 0x01 | Cmd0 = 0x2F | Cmd1 = 0x05 | Commissioning mode |

**Attributes:**

| Attribute | Length (byte) | Description |
|-----------|---------------|-------------|
| Commissioning Mode | 1 | This value specifies the commissioning mode according to this list: <br><br> 0x00    Initialization <br><br> 0x01    TouchLink <br><br> 0x02    Network Steering <br><br> 0x04    Network Formation <br><br> 0x08    Finding and Binding |

**SRSP:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x6F | Cmd1 = 0x05 | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

## 3.13.1.6  APP_CNF_BDB_SET_CHANNEL

**Description:**

Sets BDB primary or secondary channel masks.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x05 | Cmd0 = 0x2F | Cmd1 = 0x08 | isPrimary |

| 4 |
|---|
| Channel |

**Attributes:**

| Attribute | Length (byte) | Description | |
|---|---|---|---|
| isPrimary | 1 | This value specifies if the present channel set is either primary or secondary. | |
| Channel | 4 | This is the value of the channel set according to the following list of masks. | |
| | | Channel 11 | 0x00000800 |
| | | Channel 12 | 0x00001000 |
| | | Channel 13 | 0x00002000 |
| | | Channel 14 | 0x00004000 |
| | | Channel 15 | 0x00008000 |
| | | Channel 16 | 0x00010000 |
| | | Channel 17 | 0x00020000 |

| | |
|---|---|
| Channel 18 | 0x00040000 |
| Channel 19 | 0x00080000 |
| Channel 20 | 0x00100000 |
| Channel 21 | 0x00200000 |
| Channel 22 | 0x00400000 |
| Channel 23 | 0x00800000 |
| Channel 24 | 0x01000000 |
| Channel 25 | 0x02000000 |
| Channel 26 | 0x04000000 |

**SRSP:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x6F | Cmd1 = 0x08 | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.13.1.7    APP_CNF_BDB_ADD_INSTALLCODE

**Description:**

Add a preconfigured key (plain key or IC) to Trust Center device.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x19-0x1B | Cmd0 = 0x2F | Cmd1 = 0x04 | installCodeFormat |
| 8 | 16-18 | | |
| IEEEAddress | installCode | | |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| installCodeFormat | 1 | This value specifies the format in which the install code is being added. The following list contains the values |

corresponding to the supported formats:

| | |
|---|---|
| 0x01 | Install Code + CRC |
| 0x02 | Key derived from Install Code |

| | | |
|---|---|---|
| IEEEAddress | 8 | Full IEEE address for the device joining the network |
| installCode | 16-18 | 16 Bytes for the Key derived from the IC |
| | | 18 Bytes for the Install Code +CRC |

**SRSP:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x6F | Cmd1 = 0x04 | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status values: 0x00 Success 0x01 Failure (IC not supported) 0x02 Invalid parameter (bad CRC) |

### 3.13.1.8    APP_CNF_BDB_SET_TC_REQUIRE_KEY_EXCHANGE

**Description:**

Sets the policy flag on Trust Center device to mandate or not the TCLK exchange procedure.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x2F | Cmd1 = 0x09 | bdbTrustCenterRequireKeyExchange |

**Attributes:**

| Attribute | Length | Description |
|---|---|---|

| | **(byte)** | |
|---|---|---|
| bdbTrustCenterRequireKeyExchange | 1 | The bdbTrustCenterRequireKeyExchange attribute specifies whether the Trust Center requires a joining device to exchange its initial link key with a new link key generated by the Trust Center. If bdbTrustCenterRequireKeyExchange is equal to TRUE, the joining node must undergo the link key exchange procedure; failure to exchange the link key will result in the node being removed from the network. If bdbTrustCenterRequireKeyExchange is equal to FALSE, the Trust Center will permit the joining node to remain on the network without exchanging its initial link key. This attribute is used by ZigBee coordinator nodes. |

**SRSP:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x6F | Cmd1 = 0x09 | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

### 3.13.1.9    APP_CNF_BDB_SET_JOINUSESINSTALLCODEKEY

**Description:**
Sets the policy to mandate or not the usage of an Install Code upon joining.

**Usage:**
**SREQ:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x2F | Cmd1 = 0x06 | bdbJoinUsesInstallCodeKey |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| | | |

|  |  |
|---|---|
| bdbJoinUsesInstallCodeKey | 1 |

If it is equal to TRUE and the installation code derived link key is not stored, the Trust Center SHALL terminate the procedure for adding a new node into the network. If *bdbJoinUsesInstall-CodeKey* is equal to TRUE and the installation code derived link key is stored, the Trust Center SHALL first find the entry in *apsDeviceKeyPairSet* that corresponds to the joining node and then overwrite the *LinkKey* entry with the installation code derived link key and set the *KeyAttributes* field to PROVISIONAL_KEY. The Trust Center MAY then set *OutgoingFrame- Counter* to 0 and SHALL set *IncomingFrameCounter* to 0. If *bdbJoinUsesInstallCodeKey* is equal to FALSE, the Trust Center SHALL first find the entry in *apsDeviceKeyPairSet* that corresponds to the joining node and then overwrite the *LinkKey* entry with the default global Trust Center link key and set the *KeyAttributes* field to PROVISIONAL_KEY. The Trust Center MAY then set *OutgoingFrameCounter* to 0 and SHALL set *IncomingFrameCounter* to 0.

**SRSP:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x6F | Cmd1 = 0x06 | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status is either Success (0) or Failure (1). |

## 3.13.1.10  APP_CNF_BDB_SET_ACTIVE_DEFAULT_CENTRALIZED_K EY

**Description:**

On joining devices, set the default key or an install code to attempt to join the network.

**Usage:**

**SREQ:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x13 | Cmd0 = 0x2F | Cmd1 = 0x07 | CentralizedLinkKeyModes |

| 0-18 |
|---|
| InstallCode |

**Attributes:**

| Attribute | Length (byte) | | Description |
|---|---|---|---|
| CentralizedLinkKeyModes | 1 | | This parameter controls which key will be used when performing association to a centralized network. |
| | | 0x00 | Instruct joining node to use Default Global Trust Center link key. No key buffer requiered |
| | | 0x01 | Instruct the joining node to use the provided install code (16 bytes + 2 CRC bytes) to derive APS Link key to be used during joining |
| | | 0x02 | Instruct the joining node to use the provided install code (16 bytes + 2 CRC bytes) to derive APS Link key to be used during joining.  If it fails to decrypt Transport Key, it will automatically try Default Global Trust Center Link Key |
| | | 0x03 | Instruct the joining node to use the provided APS Link key to be used during joining (key size is 16 bytes) |
| | | 0x04 | Instruct the joining node to use the provided APS Link key to be used during joining (key size is 16 bytes). If it fails to decrypt Transport Key, it will automatically try Default Global Trust Center Link Key |
| InstallCode | 0-18 | | Buffer with the key in any of its formtats. |

**SRSP:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x6F | Cmd1 = 0x07 | Status |

**Attributes**:

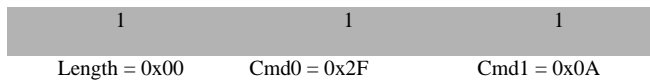| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status values: 0x00 Success 0x01 Failure (IC not supported) 0x02 Invalid Parameters (bad CRC) |

### 3.13.1.11   APP_CNF_BDB_ZED_ATTEMPT_RECOVER_NWK

**Description:**

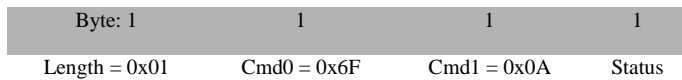Instruct the ZED to try to rejoin its previews network. Use only in ZED devices.

**Usage:**

**SREQ:**

| 1 | 1 | 1 |
|---|---|---|
| Length = 0x00 | Cmd0 = 0x2F | Cmd1 = 0x0A |

**Attributes:**

**SRSP:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x6F | Cmd1 = 0x0A | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status values:<br><br>0x00 Success<br><br>0x01 Failure (Device not ZED or doesn't have network parameters) |

### 3.13.2   MT_APP_CONFIG Callbacks

### 3.13.2.1   APP_CNF_BDB_COMMISSIONING_NOTIFICATION

**Description:**

Callback to receive notifications from BDB process.

**Usage:**

**AREQ:**

| Byte: 1 | 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|---|
| Length = 0x03 | Cmd0 = 0x4F | Cmd1 = 0x80 | Status | Commissioning mode | Remaining commissioning modes |

**Attributes:**

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status of the commissioning mode being notified:<br><br>0x00  BDB_COMMISSIONING_SUCCESS<br><br>0x01  BDB_COMMISSIONING_IN_PROGRESS<br><br>0x02  BDB_COMMISSIONING_NO_NETWORK<br><br>0x03  BDB_COMMISSIONING_TL_TARGET_FAILURE<br><br>0x04  BDB_COMMISSIONING_TL_NOT_AA_CAPABLE<br><br>0x05  BDB_COMMISSIONING_TL_NO_SCAN_RESPONSE<br><br>0x06  BDB_COMMISSIONING_TL_NOT_PERMITTED<br><br>0x07  BDB_COMMISSIONING_TCLK_EX_FAILURE<br><br>0x08  BDB_COMMISSIONING_FORMATION_FAILURE<br><br>0x09  BDB_COMMISSIONING_FB_TARGET_IN_PROGRESS<br><br>0x0A BDB_COMMISSIONING_FB_INITIATOR_IN_PROGRESS<br><br>0x0B BDB_COMMISSIONING_FB_NO_IDENTIFY_QUERY_RESPONSE<br><br>0x0C  BDB_COMMISSIONING_FB_BINDING_TABLE_FULL<br><br>0x0D  BDB_COMMISSIONING_NETWORK_RESTORED<br><br>0x0E  BDB_COMMISSIONING_FAILURE |
| Commissioning mode | 1 | Commissioning mode for which the notification is done and to which the status is related:<br><br>0x00  BDB_COMMISSIONING_INITIALIZATION |

0x01  BDB_COMMISSIONING_NWK_STEERING

0x02  BDB_COMMISSIONING_FORMATION

0x03  BDB_COMMISSIONING_FINDING_BINDING

0x04  BDB_COMMISSIONING_TOUCHLINK

0x05  BDB_COMMISSIONING_PARENT_LOST

Bitmask of the remaining commissioning modes after this notification:

0x01 BDB_COMMISSIONING_MODE_INITIATOR_TL

0x02 BDB_COMMISSIONING_MODE_NWK_STEERING

| Remaining commissioning modes | 1 | 0x04 BDB_COMMISSIONING_MODE_NWK_FORMATION |

0x08 BDB_COMMISSIONING_MODE_FINDING_BINDING

0x10 BDB_COMMISSIONING_MODE_INITIALIZATION

0x20 BDB_COMMISSIONING_MODE_PARENT_LOST

## 3.14    MT_GREENPOWER

This interface allows a host processor to interact with Green Power devices and Green Power infrastructure.

### 3.14.1  MT_GP Commands

#### 3.14.1.1    GP_DATA_REQ

**Description:**

Callback to receive notifications from BDB process.

**Usage:**

**SREQ**

| Byte: 1 | 1 | 1 | 1 | 1 | 1 | 4 |
|---|---|---|---|---|---|---|
| Length = 0x17-0x8C | Cmd0 = 0x35 | Cmd1 = 0x01 | Action | TxOptions | Application ID | SrcID |

| 8 | 1 | 1 | 1 | 0-116 | 1 | 3 |
|---|---|---|---|---|---|---|
| GPD IEEE address | EndPoint | GPD Command ID | GPD ASU Length | GPD ASU | GPEP handle | gpTxQueueEntry-Lifetime |

**Attributes:**

| Attribute | Length (byte) | Range | Description |
|---|---|---|---|
| Action | 1 | TRUE/FALSE | TRUE: add GPDF into the queue<br><br>FALSE: remove GPDF from queue |
| TxOptions | 1 | AnyValid | The transmission options for this GPDF. These are a bitwise OR of one or more of the following:<br><br>b0 = Use gpTxQueue<br><br>b1 = Use CSMA/CA<br><br>b2 = Use MAC ACK<br><br>b3-b4 = GPDF frame type for Tx (can take non-reserved values as defined in Table 10)<br><br>b5 = Tx on matching endpoint<br><br>b6 – b7 – reserved |
| Application ID | 1 | 0x00, 0x02 | ApplicationID of the GPD to which the ASDU will be sent;<br><br>ApplicationID 0x00 indicates the usage of the SrcID;<br><br>ApplicationID 0x02 indicates the usage of the GPD IEEE address. |
| SrcID | 4 | 0x00000000 - 0xffffffff | The identifier of the GPD entity to which the ASDU will be sent if ApplicationID = 0b000. |
| GPD IEEE address | 8 | Any valid | The identifier of the GPD entity to which the ASDU will be sent if ApplicationID = 0b010. |
| Endpoint | 1 | 0x00 - 0xf0, 0xff | The identifier of the GPD endpoint used in combination with the GPD IEEE address if ApplicationID = 0b010.<br><br>If ApplicationID = 0b000 this parameter is ignored. |
| GPD Command ID | 1 | 0x00- 0xff | The identifier of the command, within the GP specification, which defines the application semantics of the ASDU. |
| GPD ASDU Length | 1 | 0x00 - (aMaxMACFrameSize - 9) | The number of octets in the transmitted GPD ASDU. |
| GPD ASDU | GPD ASU Length | - | The sequence of octets forming the transmitted GPD ASDU. |

|  |  |  |  |
|---|---|---|---|
| GPEP handle | 1 | 0x00-0xff | The handle used between Green Power EndPoint and dGP stub, to match the request with the confirmation. |
| gpTxQueueEntry-Lifetime | 3 | 0x000000-0xffffff | The lifetime of this packet in the gpTxQueue, in milliseconds. 0x000000 indicates immediate transmission. 0xffffff indicates infinity. In a Basic Proxy/Sink, the default lifetime MAY be 0xffffff. |

**SRSP:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x75 | Cmd1 = 0x01 | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status values: 0x00 Success 0x01 Failure (Out of memory) 0x02 Invalid Parameters (Invalid App ID) |

### 3.14.1.2    GP_SEC_RSP

**Description:**

This message provides a mechanism for the Green Power EndPoint to provide security data into the dGP stub.

**Usage:**

**SREQ**

| Byte: 1 | 1 | 1 | 1 | 1 | 1 | 4 |
|---|---|---|---|---|---|---|
| Length = 0x26 | Cmd0 = 0x35 | Cmd1 = 0x02 | Status | dGP stub handle | ApplicationID | SrcID |

| 8 | 1 | 1 | 1 | 16 | 4 |
|---|---|---|---|---|---|
| GPD IEEE address | EndPoint | GPDFScurityLevel | GPDFKeyType | GPDKey | GPDSecurityFrameCounter |

**Attributes:**

| Attribute | Length (byte) | Range | Description |
|---|---|---|---|
| Status | 1 | Any valid | The status code, as returned by the Green Power EndPoint. The following are supported: MATCH DROP_FRAME PASS_UNPROCESSED TX_THEN_DROP |
| dGP stub handle | 1 | 0x00-0xff | The handle used between dGP stub and the higher layers, to match the request with the response. |
| Application ID | 1 | 0x00, 0x02 | ApplicationID of the GPD entity from which the ASDU was received. ApplicationID 0x00 indicates the usage of the SrcID; ApplicationID 0x02 indicates the usage of the GPD IEEE address. |
| SrcID | 4 | 0x00000001 - 0xfffffffe | The identifier of the GPD entity from which the ASDU was received if ApplicationID = 0b000. |
| GPD IEEE address | 8 | Any valid | The identifier of the GPD entity from which the ASDU was received if ApplicationID = 0b010. |
| Endpoint | 1 | 0x00 - 0xf0, 0xff | The identifier of the GPD endpoint used in combination with the GPD IEEE address if ApplicationID = 0b010. If ApplicationID = 0b000 this parameter is ignored. |
| GPDFSecurityLevel | 1 | 0x00 - 0x07 | The security level to be used for GPDF security processing. |
| GPDFKeyType | 1 | Any valid | The security key type to be used for GPDF security processing. |
| GPD Key | 16 | Any valid | The security key to be used for GPDF security processing. |
| GPD security frame counter | 4 | Any valid | The security frame counter value to be used for GPDF security processing. |

**SRSP:**

| Byte: 1 | 1 | 1 | 1 |
|---|---|---|---|
| Length = 0x01 | Cmd0 = 0x75 | Cmd1 = 0x02 | Status |

**Attributes**:

| Attribute | Length (byte) | Description |
|---|---|---|
| Status | 1 | Status values:<br><br>0x00 Success<br><br>0x01 Failure (Out of memory)<br><br>0x02 Invalid Parameters (Invalid App ID) |

## 3.14.2 MT_GP Callbacks

### 3.14.2.1 GP_DATA_CNF

**Description:**

Green power confirm is a message that provides a mechanism for the Green Power EndPoint in the host processor to understand the status of a previous request to send a GPDF.

**Usage:**

**AREQ**

| Byte: 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|
| Length = 0x02 | Cmd0 = 0x55 | Cmd1 = 0x05 | Status | GP MPDU handle |

**Attributes:**

| Attribute | Length (byte) | Range | Description |
|---|---|---|---|
| Status | 1 | Any valid | Status code, as returned by the MAC layer (see Table 28 of [15]). |
| GP MPDU | 1 | 0x00 - 0xff | The handle used between dGP/dLPED stub and cGP stub, to match the request with the confirmation. |

### 3.14.2.2    GP_SEC_REQ

**Description:**

This message provides a mechanism for dGP stub to request security data from the Green Power EndPoint in the host processor.

**Usage:**

**AREQ**

| Byte: 1 | 1 | 1 | 1 | 4 | 8 |
|---|---|---|---|---|---|
| Length = 0x12-0x15 | Cmd0 = 0x53 | Cmd1 = 0x03 | ApplicationID | SrcID | GPD IEEE address |

| 1 | 1 | 1 | 1, 4 | 1 |
|---|---|---|---|---|
| Endpoint | GPDFScurityLevel | GPDFKeyType | GPD Security Frame Counter | dGP stub handle |

**Attributes:**

| Attribute | Length (byte) | Range | Description |
|---|---|---|---|
| ApplicationID | 1 | 0x00, 0x02 | ApplicationID of the GPD entity from which the ASDU was received.<br><br>ApplicationID 0x00 indicates the usage of the SrcID; ApplicationID 0x02 indicates the usage of the GPD IEEE address. |
| SrcID | 4 | 0x00000001 - 0xfffffffe | The identifier of the GPD entity from which the ASDU was received if ApplicationID = 0b000. |
| GPD IEEE address | 8 | Any valid | The identifier of the GPD entity from which the ASDU was received if ApplicationID = 0b010. |
| EndPoint | 1 | 0x00 - 0xf0, 0xff | The identifier of the GPD endpoint used in combination with the GPD IEEE address if ApplicationID = 0b010.<br>If ApplicationID = 0b000 this parameter is ignored. |
| GPDFSecurity Level | 1 | 0x00, 0x02 - 0x03 | The security level, corresponding to the received MPDU. |
| GPDFKey Type | 1 | 0x00 - 0x01 | The security key type, corresponding to the received MPDU. |
| GPD Security Frame Counter | 1, 4 | As specified by the GPDF Security Level parameter | The security frame counter value corresponding to the received MPDU. |
| dGP stub handle | 1 | 0x00-0xff | The handle used between dGP stub and the higher layers, to match the request with the response. |

### 3.14.2.3    GP_DATA_IND

**Description:**

This message provides a mechanism for identifying and conveying a received GPDF to the Green Power EndPoint in the host processor.

**Usage:**

**AREQ**

| Byte: 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|
| Length = 0x24-0x91 | Cmd0 = 0x55 | Cmd1 = 0x04 | Status | RSSI | LinkQuality | SeqNumber | SrcAddrMode |

| 2 | 8 | 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|
| SrcPANID | SrcAddress | frameType | appID | GPDFSecLvl | GPDFKeyType | Auto Commissioning |

| 1 | 4 | 1 | 4 | 1 | 1 | - |
|---|---|---|---|---|---|---|
| RxAfterTx | SrcId | EndPoint | GPDSecFrame Counter | GPDCmdID | GPDasduLength | GPDasdu |

| 4 |
|---|
| MIC |

**Attributes:**

| Attribute | Length (byte) | Range | Description |
|---|---|---|---|
| Status | 1 | Any Valid | Status code, as returned by dGP stub. It can have the following values: SECURITY_SUCCESS NO_SECURITY COUNTER_FAILURE AUTH_FAILURE UNPROCESSED |
| RSSI | 1 | 0x00 - 0xff | The RSSI delivered by the MAC on receipt of this frame. |
| Link Quality | 1 | 0x00 - 0xff | The LQI delivered by the MAC on receipt of this frame. |
| SeqNumber | 1 | 0x00 - 0xff | The sequence number from MAC header of the received MPDU. |
| SrcAddrMode | 1 | 0x00 - 0x03 | The source addressing mode for this primitive corresponding to the received MPDU. This value can take one of the following values: 0 x 00 = no address (SrcPANId and SrcAddress omitted). 0 x 01 = reserved. 0 x 02 = 16 bit short address. 0 x 03 = 64 bit extended address. |

| | | | |
|---|---|---|---|
| SrcPANId | 2 | 0x0000 - 0xffff | The 16-bit PAN identifier of the GPD entity from which the ASDU was received. |
| SrcAddress | 8 | As specified by the SrcAddrMode parameter | The device address of the GPD entity from which the ASDU was received. Always sent as uint64. |
| frameType | 1 | 0x00, 0x01 | The number of octets in the received GP MPDU. |
| appID | 1 | 0x00, 0x02 | Application ID of the GPD entity from which the ASDU was received. |
| GPDFSecLvl | 1 | 0x00, 0x02 - 0x03 | The security level, corresponding to the received MPDU. |
| GPDFKeyType | 1 | 0x00 - 0x01 | The security key type, corresponding to the received MPDU. |
| Auto Commissioning | 1 | 0x00, 0x01 | Auto commissioning value received from GPD. |
| RxAfterTx | 1 | 0x00, 0x01 | If the value of this field is 0x01, then it indicates that the GPD will enter the receive mode. |
| SrcId | 4 | 0x00000001 - 0xfffffffe | The identifier of the GPD entity from which the ASDU was received if ApplicationID = 0b000. |
| EndPoint | 1 | 0x00 - 0xf0, 0xff | The identifier of the GPD endpoint used in combination with the GPD IEEE address if ApplicationID = 0b010. If ApplicationID = 0b000 this parameter is ignored. |
| GPDSecFrame Counter | 4 | Any valid | The security frame counter value to be used for GPDF security processing. |
| GPDCmdID | 1 | 0x00- 0xff | The identifier of the command, within the GP specification, which defines the application semantics of the ASDU. |
| GPDasduLength | 1 | 0x00 - (aMaxMACFrameSize - 9) | The number of octets in the transmitted GPD ASDU. |
| GPDasdu | GPD ASU Length | - | The sequence of octets forming the transmitted GPD ASDU. |
| MIC | 4 | Any valid | The sequence of octets forming the MIC for the received GPD MPDU. |