Chipcon Products from Texas Instruments

Z-Stack Compile Options

Document Number: F8W-2005-0038

Texas Instruments, Inc.
San Diego, California USA
(619) 497-3845

Version	Description	Date
1.0	Initial release.	12/22/2005
1.1	Added FORCE_MAC_NEAR and MINIMIZE_ROOT options.	03/08/2006
2.0	Release 1.4.0.	12/07/2006
2.1	Updated description of the ASSERT_RESET and SECURE options.	04/05/2007

Table of Contents

1. Introduction

1.1. Scope

This document provides information and procedures for using compiler options with Texas Instruments' Z-Stack.

2. Requirements

2.1. Target Development System Requirements

Z-Stack provides a complementary offering to the IAR Embedded Workbench (EW8051) suite of software development tools. These tools support project management, compiling, assembling, linking, downloading, and debugging for various 8051-based processors, including the Chipcon CC2430 family. The following is required support for the Z-Stack target development system:

- IAR EW8051 (http://www.iar.com/)

3. Using Z-Stack Compile Options

3.1. Selecting the Logical Device Type

ZigBee devices can be configured in one of three ways:

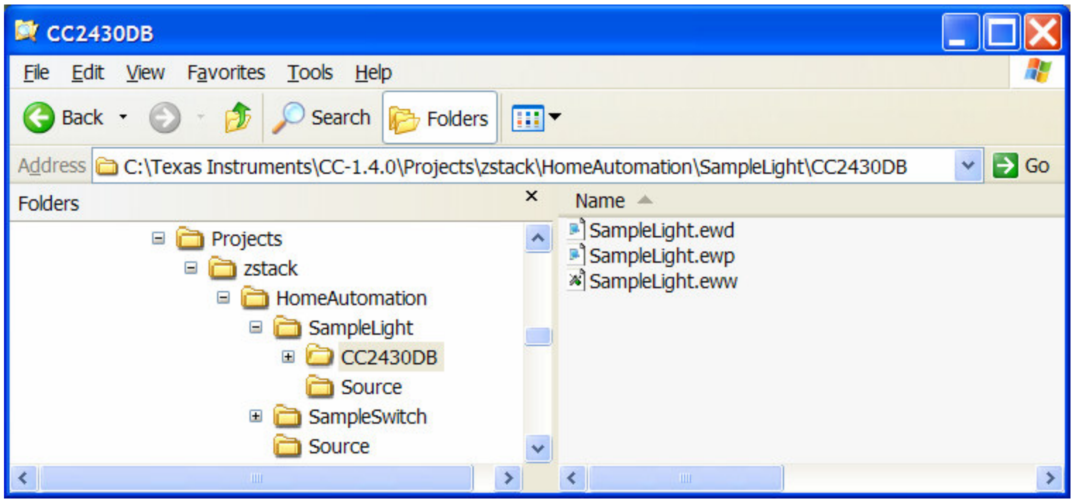
- ZigBee Coordinator – This device is configured to start the IEEE 802.15.4 network and will serve as the PAN Coordinator in that network.
- ZigBee Router – This device is configured to associate with a ZigBee Coordinator, then allow other routers or end devices to associate with it. It will route data packets in the network.
- ZigBee End Device – This device is configured to join a pre-existing network and will associate with a ZigBee Coordinator or ZigBee Router.

3.2. Locating Compile Options

Compile options for a specific project are located in two places. Options that are rarely, if ever, changed are located in linker control files, one for each logical device type discussed above. User-defined options and ones that change to enable/disable features are located in the IAR project file. For demonstration purposes, these two files for the SampleLight Coordinator project will be examined. Access to all other Z-Stack projects will be similar.

3.3. Compile Options In Linker Control Files

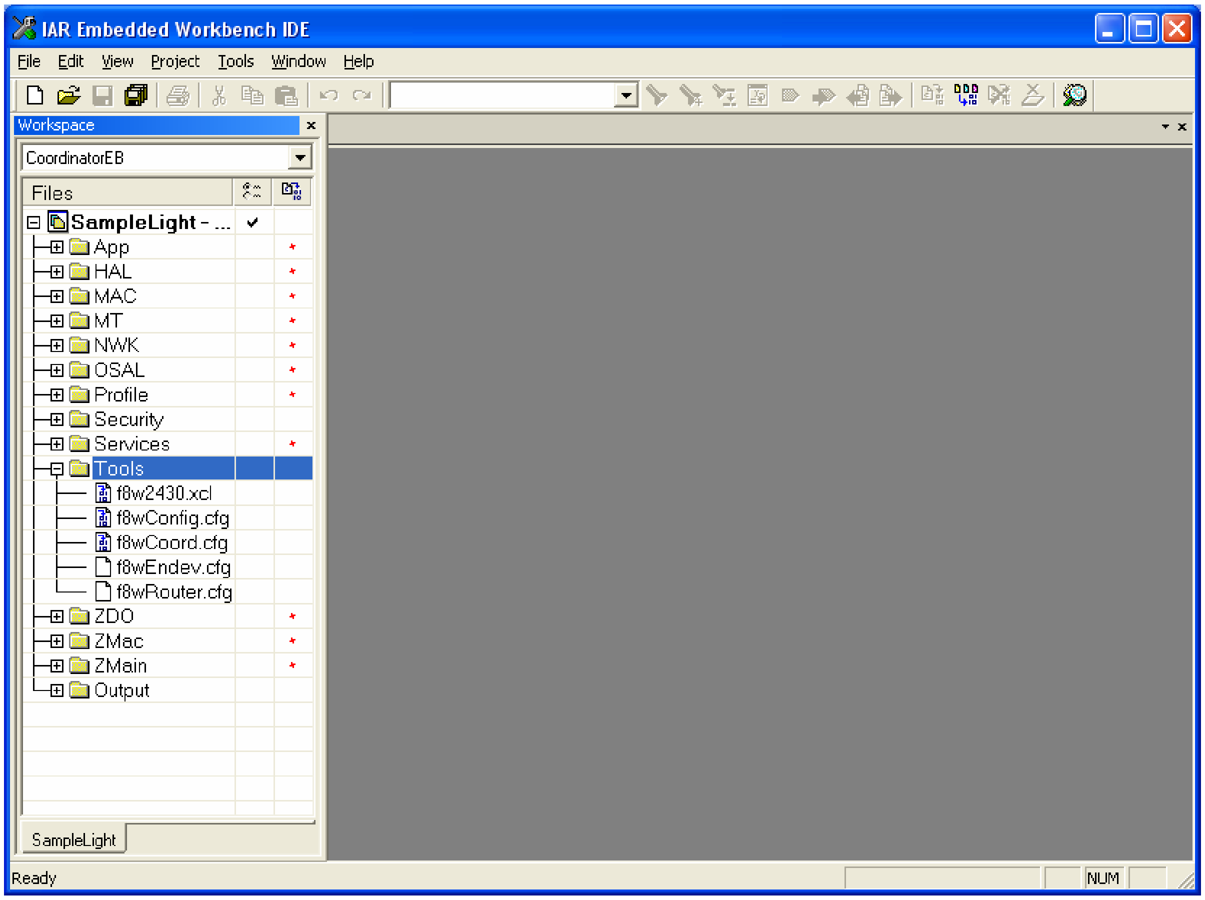
SampleLight project files are found in the **..\Projects\zstack\HomeAutomation\SampleLight\CC2430DB** folder:



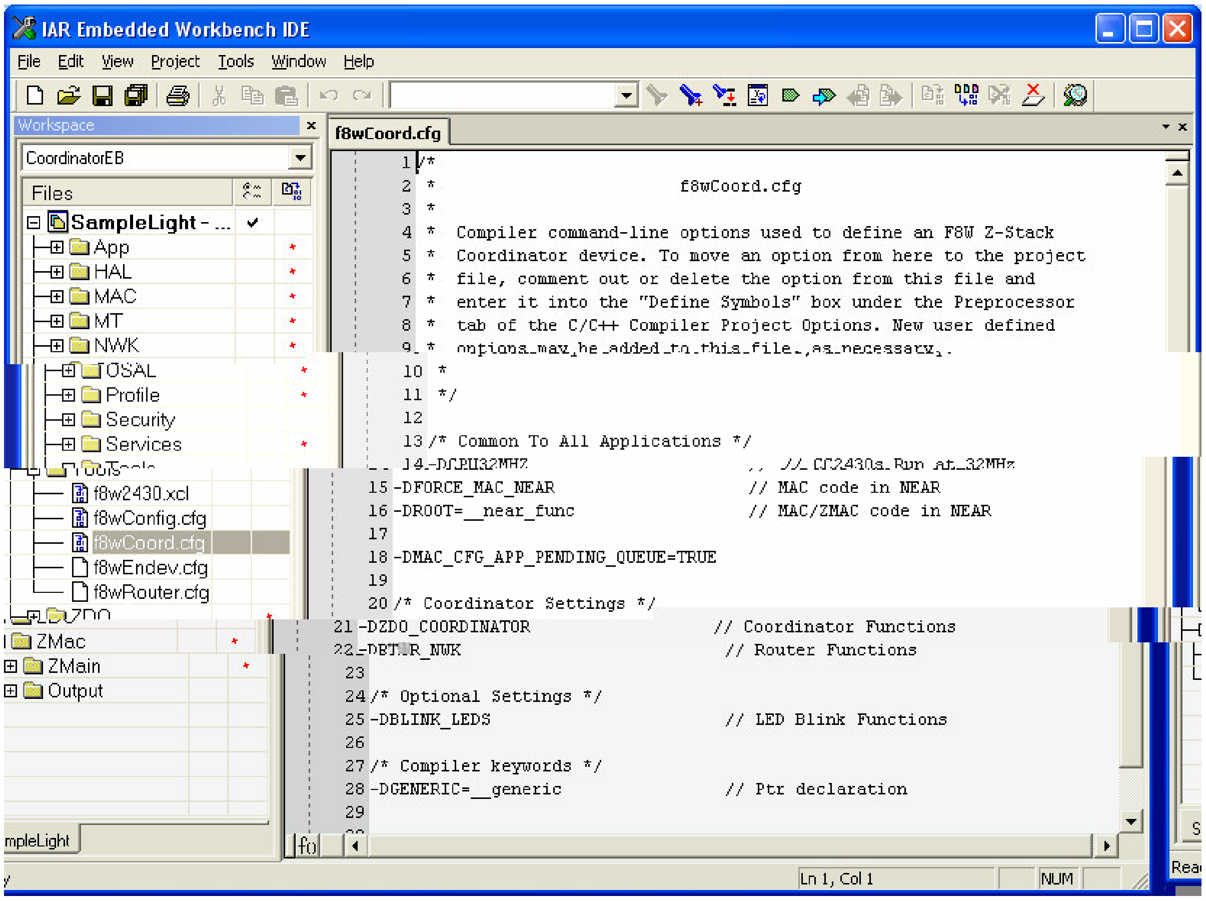
C:\Texas Instruments\CC-1.4.0\Projects\zstack\HomeAutomation\SampleLight\CC2430DB

SampleLight.ewd
SampleLight.ewp
SampleLight.eww

Open the project by double-clicking on the *SampleLight.eww* file, select the *CoordinatorEB* configuration from the pull-down list below **Workspace**, and then open the **Tools** folder. Several linker control files are located in the **Tools** folder. This folder contains various configuration files and executable tools used in Z-Stack projects. Generic compile options are defined in the *f8wConfig.cfg* file. This file, for example, specifies the channel(s) and the PAN ID that will be used when a device starts up. This is the recommended location for a user to establish specific channel settings for their projects. This allows developers set up "personal" channels to avoid conflict with others. Device specific compile options are located in the *f8wCoord.cfg*, *f8wEndev.cfg*, and *f8wRouter.cfg* files:



The SampleLight Coordinator project uses the *f8wCoord.cfg* file. As shown below, compile options that are specific to Coordinator devices and options that provide “generic” Z-Stack functions are included in this file:



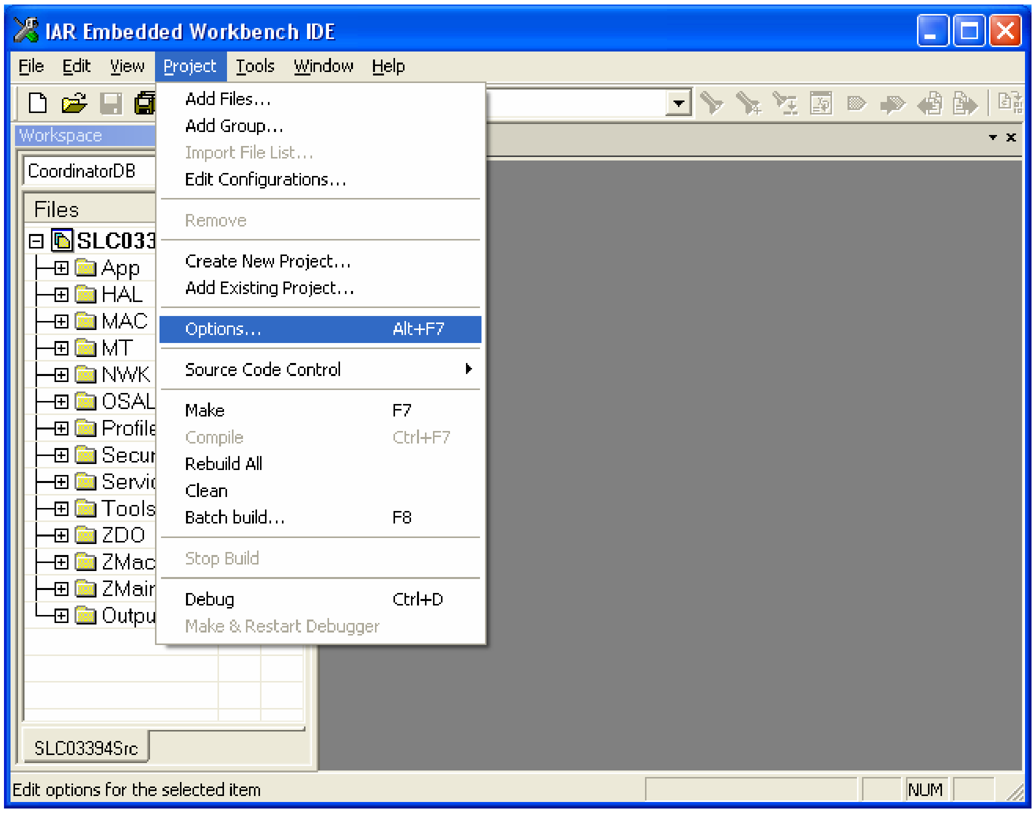
```
/*
 *                         f8wCoord.cfg
 *
 *  Compiler command-line options used to define an F8W Z-Stack
 *  Coordinator device. To move an option from here to the project
 *  file, comment out or delete the option from this file and
 *  enter it into the "Define Symbols" box under the Preprocessor
 *  tab of the C/C++ Compiler Project Options. New user defined
 *  options may be added to this file, as necessary.
 *
 */

/* Common To All Applications */
-DCPU32MHZ                          // CC2430s Run at 32MHz
-DFORCE_MAC_NEAR                    // MAC code in NEAR
-DROOT=__near_func                  // MAC/ZMAC code in NEAR

-DMAC_CFG_APP_PENDING_QUEUE=TRUE

/* Coordinator Settings */
-DZDO_COORDINATOR                   // Coordinator Functions
-DRTR_NWK                           // Router Functions

/* Optional Settings */
-DBLINK_LEDS                        // LED Blink Functions

/* Compiler keywords */
-DGENERIC=__generic                 // Ptr declaration
```

The *f8wCoord.cfg* file is used by all projects that build Coordinator devices. Therefore, any change made to this file will affect all Coordinators. In a similar manner, the *f8wRouter.cfg* and *f8wEnd.cfg* files affect all Router and End-Device projects, respectively.

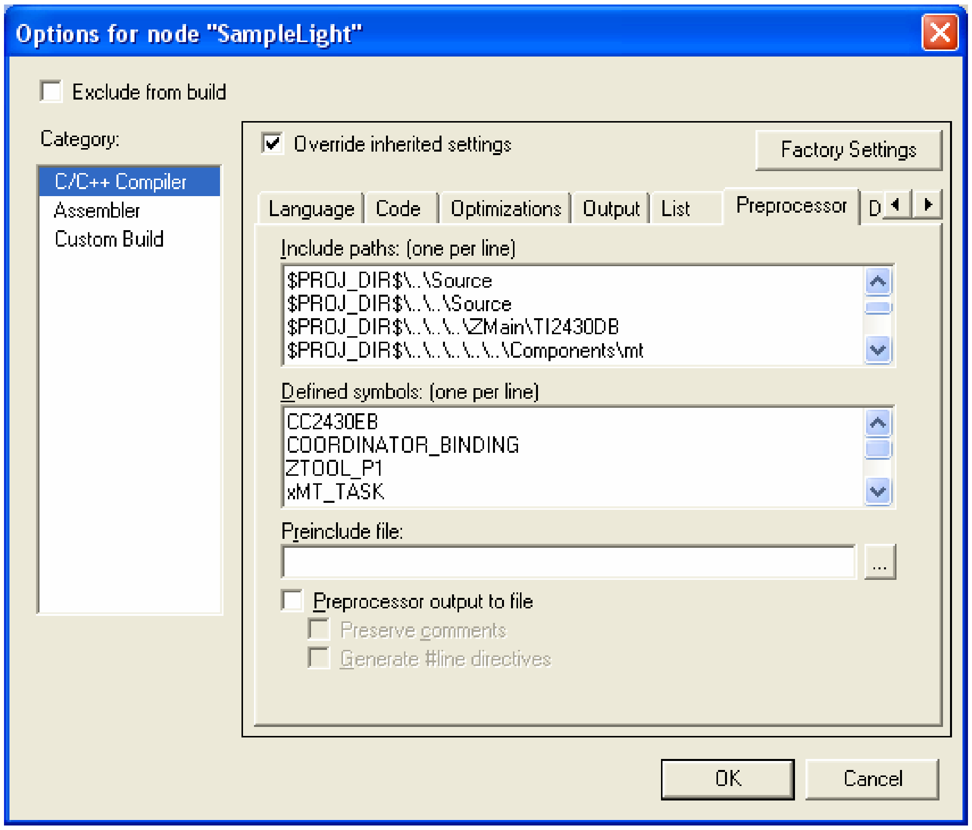
To add a compile option to all projects of a certain device type, simply add a new line to the appropriate linker control file. To disable a compile option, comment that option out by placing // at the left edge of the line. You could also delete the line but this is not recommended since the option might need to be re-enabled at a later time.

3.4. Compile Options In IAR Project Files

The compile options for each of the supported configurations are stored in the *SampleLight.ewp* file. To modify the compile options, select the **Options…** item from the **Project** pull-down menu:



Select the **C/C++ Compiler** item and click on the **Preprocessor** tab. The compile options for this configuration are located in the box labeled *Defined symbols: (one per line)*:



Include paths: (one per line)

```
$PROJ_DIR$\..\Source
$PROJ_DIR$\..\..\Source
$PROJ_DIR$\..\..\..\ZMain\TI2430DB
$PROJ_DIR$\..\..\..\..\..\Components\mt
```

Defined symbols: (one per line)

```
CC2430EB
COORDINATOR_BINDING
ZTOOL_P1
xMT_TASK
```

To add a compile option to this configuration, simply add the item on a new line within this box. To disable a compile option, place an x at the left edge of the line. Note that the **MT_ZDO_FUNC** option has been disabled in the example shown above. This option could have been deleted but this is not recommended since it might need to be re-enabled at a later time.

3.5. Using Compile Options

Compile options are used to select between features that are provided in the source files. Most compile options act as on/off switches for specific sections within source programs. Some options are used to provide a user-defined numerical value, such as *DEFAULT_CHANLIST*, to the compiler to override default values.

Each of the Z-Stack projects provides an IAR project file which specifies the compile options to be used for that specific project. The programmer can add or remove options as needed to include or exclude portions of the available software functions. Note that changing compile options may require other changes to the project file. For example, adding the *MT_NWK* options requires *MT_NWK.c* to be in the list of source files and the use of the appropriate MT-enabled network library - if you are changing the SampleLight Coordinator project, which normally uses the *Router.lib* file, the *RouterMt.lib* file must be used instead.

The next sections of this document provide lists of the supported compile options with a brief description of what feature they enable or disable. Options that are listed as "*do not change*" are required for proper operation of the compiled programs. Options that are listed as *"do not use"* are not appropriate for use with the CC2430 boards.

4. Supported Compile Options and Definitions

4.1. General Compile Options

The compile options in the following table can be changed or set:

APS_DEFAULT_INTERFRAME_DELAY	Delay between Tx packets when using fragmentation (not yet supported)
APS_DEFAULT_MAXBINDING_TIME	Maximum time in seconds that a Coordinator will wait between receiving match descriptor bind requests to perform binding
APS_DEFAULT_WINDOW_SIZE	Size of a Tx window when using fragmentation (not yet supported)
APS_MAX_GROUPS	Maximum number of groups in the groups table
APS_MAX_GROUPS	Maximum number of entries allowed in the groups table
APSC_ACK_WAIT_DURATION_POLLED	Number of 2 milliseconds periods a polling End Device will wait for an APS acknowledgement from the destination device
APSC_MAX_FRAME_RETRIES	Maximum number of retries allowed (at APS layer) after a transmission failure
ASSERT_RESET	Specifies that the device should reset when there's an assertion. When not defined, all LEDs will flash when an assertion occurs.
BEACON_REQUEST_DELAY	Minimum number of milliseconds to delay between each beacon request in a joining cycle
BLINK_LEDS	Enable extended LED blinking functions
COORDINATOR_BINDING	Enable Coordinator binding (Coordinator only)
DEF_PROTO_VERS	Set to 1 or 2 for ZigBee Protocol Version 1.0 or 1.1, respectively. Forces Coordinator to start only specified network version and joining device to join only the specified version network.
DEFAULT_CHANLIST	Override the default channel definition in file NLMEDE.h
DEFAULT_KEY	Default security key
ED_BIND	Enable bind/unbind processing when COORDINATOR_BINDING not active
EXTENDED_JOINING_RANDOM_MASK	Mask for the random joining delay
HOLD_AUTO_START	Disable automatic start-up of ZDApp event processing loop
KB_INT	Enable keyboard (joystick) interrupt
KEYPOLL	Enable key-polling
LCD_SUPPORTED	Enable LCD emulation – text sent to ZTool serial port. Optional parameter =DEBUG copies LCD messages to the debug port
MAC_CFG_APP_PENDING_QUEUE	MAC Settings
MAC_OPT_FFD	Enable Full Function Device (FFD)
MANAGED_SCAN	Enable delays between channel scans
MAX_BCAST	Maximum number of simultaneous broadcasts supported by a device at any given time
MAX_BINDING_CLUSTER_IDS	Maximum number of cluster IDs for each binding table entry
MAX_BINDING_CLUSTER_IDS	Maximum number of cluster IDs in a binding record
MAX_POLL_FAILURE_RETRIES	Number of times retry to poll parent before indicating loss of synchronization with parent. Note that larger value will cause longer delay for the child to rejoin the network
MAX_RREQ_ENTRIES	Number of simultaneous route discoveries in network
MAX_RREQ_ENTRIES	Maximum number of RREQ packets in the network
MAX_RTG_ENTRIES	Number of entries in the regular routing table plus additional entries for route repair
MAX_RTG_ENTRIES	Number of entries in the regular routing table
MAXMEMHEAP	Determines the total memory available for dynamic memory. Every request for an amount of dynamic memory requires dynamic memory space for overhead used in managing the allocated memory. So MAXMEMHEAP does not reflect the total amount of dynamic memory that the user can expect to be usable. As a rule of thumb, each memory allocation requires at least 2+N bytes, where N represents the word-alignment block size of the target CPU (e.g. N=1 on the AVR and CC2430 but N=2 on the MSP430.) MAXMEMHEAP must be defined to be less than 32768.

MIN_GAP	Define minimum time between transmitted serial messages
MINIMIZE_ROOT	Minimizes amount of Stack code that gets placed in the ROOT memory segment
NONWK	Disable NWK, APS, and ZDO functionality
NV_INIT	Enable loading of "basic" NV items at device reset
NV_RESTORE	Enables device to save/restore network state information to/from NV
NWK_AUTO_POLL	Enable End Device to poll from the parents automatically
NWK_INDIRECT_MSG_TIMEOUT	Number of milliseconds the parent of a polling End Device will hold a message
NWK_MAX_BINDING_ENTRIES	Maximum number of entries in the binding table
NWK_MAX_DATA_RETRIES	The maximum number of times retry looking for the next hop address of a message
NWK_MAX_DEVICE_LIST	Maximum number of devices in the Association/Device list
NWK_MAX_DEVICES	Maximum number of devices in the network
NWK_START_DELAY	Minimum number of milliseconds to hold off the start of the device in the network and the minimum delay between joining cycles
OSAL_TIMER_16_BIT	OSAL timer: FALSE = HAL_TIMER_0 (8-bit) or TRUE = HAL_TIMER_3 (16-bit)
OSAL_TOTAL_MEM	Track OSAL memory heap usage (display if LCD_SUPPORTED)
POLL_RATE	Number of milliseconds to wait between data request polls to the coordinator
POWER_SAVING	Enable power saving functions for battery-powered devices
QUEUED_POLL_RATE	This is used after receiving a data indication to poll immediately for queued messages (in milliseconds)
REFLECTOR	Enable binding
REJOIN_POLL_RATE	This is used as an alternate response poll rate only for rejoin request. This rate is determined by the response time of the parent that the device is trying to join
RESPONSE_POLL_RATE	This is used after receiving a data confirmation to poll immediately for response messages (in milliseconds)
ROUTE_EXPIRY_TIME	Number of seconds before an entry expires in the routing table; set to 0 to turn off route expiry
RTR_NWK	Enable Router networking
SECURE	Enable ZigBee security (SECURE=0 to disable, SECURE=1 to enable)
SERIAL_DTE	Define serial port as a DTE device instead of DCE device
SERIAL_RX_INT	Enable use of serial receive interrupts
SERIAL_TX_INT	Enable use of serial transmit interrupts
SERIAL_XFER	Enable serial port for non-ZTool messages
SOFT_START	Enable device to start as Coordinator if none present, otherwise become Router
TIMER_INT	Enable use of timer interrupts
USE_KEY_EXPANSION	Enable faster AES operation by using extra RAM
ZAPP_Px	Enable ZApp messages via serial port Px where x is the port (1 or 2)
ZDAPP_CONFIG_PAN_ID	Coordinator's PAN ID; used by Routers and End Devices to join PAN with this ID
ZDO_COORDINATOR	Enable the device as a Coordinator
ZTOOL_Px	Enable ZTool messages via serial port Px where x is the port (1 or 2)

The compile options in the following table cannot be changed or used:

CC2430BB	Target is a SoC-BB battery board ***(do not change)***
CC2430DB	Target is a CC2430DB evaluation board ***(do not change)***
CC2430EB	Target is a SmartRF04EB evaluation board ***(do not change)***
CPU32MHZ	Clock rate of the CPU - Can be 16 or 32 MHZ ***(do not change)***
EXTERNAL_RAM	Enable use of external RAM memory for the OSAL heap ***(do not use)***
FORCE_MAC_NEAR	Forces MAC code into the NEAR memory segment ***(do not change)***
GENERIC=__generic	Defines complier keyword for generic pointers ***(do not change)***
MACSIM	Enable MAC simulation ***(do not use)***

NWK_TEST	Enable Network test functions ***(do not use)***
ROOT=__near_func	Defines compiler keyword for ROOT memory ***(do not change)***
WIN32	Enable Windows simulation ***(do not use)***

4.2. Monitor-Test (MT) Compile Options

You can enable the following APIs and function associated with the MT_TASK option, but you must include the MT_TASK option.

MT_TASK	Enable Monitor-Test task
MT_AF_FUNC	Enable Monitor-Test processing of AF commands issued from ZTool or ZTrace
MT_AF_CB_FUNC	Enable Monitor-Test processing of AF callbacks registered by ZTool or ZTrace
MT_MAC_FUNC	Enable Monitor-Test processing of MAC commands issued from ZTool or ZTrace
MT_MAC_CB_FUNC	Enable Monitor-Test processing of MAC callbacks registered by ZTool or ZTrace
MT_NWK_FUNC	Enable Monitor-Test processing of NWK commands issued from ZTool or ZTrace
MT_NWK_CB_FUNC	Enable Monitor-Test processing of NWK callbacks registered by ZTool or ZTrace
MT_ZDO_FUNC	Enable Monitor-Test processing of ZDO commands issued from ZTool or ZTrace
MT_ZDO_MGMT	Enable Monitor-Test processing of ZDO MGMT commands from ZTool or ZTrace
MT_USER_TEST_FUNC	Enable Monitor-Test processing of User commands issued from ZTool or ZTrace
MT_NWK_PING	Enable Monitor-Test network ping between devices ***(do not use)***
MT_APS_CB_FUNC	Enable Monitor-Test processing of APS callbacks registered by ZTool or ZTrace
MT_GOF_FUNC	Enable Monitor-Test processing of GOF commands issued from ZTool or ZTrace
MT_GOF_CB_FUNC	Enable Monitor-Test processing of GOF callbacks registered by ZTool or ZTrace

4.3. ZigBee Device Object (ZDO) Compile Options

By default, the mandatory messages (as defined by the ZigBee spec) are enabled in the ZDO. All other message processing is controlled by compile flags. You can enable/disable the options by commenting/uncommenting the compile flags in ZDConfig.h or include/exclude them like other compile flags. There's an easier way to enable all the ZDO Function and Management options. You can use MT_ZDO_FUNC to enable all the ZDO Function options, and MT_ZDO_FUNC and MT_ZDO_MGMT to enable all the ZDO Function plus Management options.

ZDO_NWKADDR_REQUEST	Enable Network Address Request function and response processing
ZDO_IEEEADDR_REQUEST	Enable IEEE Address Request function and response processing
ZDO_MATCH_REQUEST	Enable Match Descriptor Request function and response processing
ZDO_NODEDESC_REQUEST	Enable Node Descriptor Request function and response processing
ZDO_POWERDESC_REQUEST	Enable Power Descriptor Request function and response processing
ZDO_SIMPLEDESC_REQUEST	Enable Simple Descriptor Request function and response processing
ZDO_ACTIVEEP_REQUEST	Enable Active Endpoint Request function and response processing
ZDO_COMPLEXDESC_REQUEST	Enable Complex Descriptor Request function and response processing
ZDO_USERDESC_REQUEST	Enable User Descriptor Request function and response processing
ZDO_USERDESCSET_REQUEST	Enable User Descriptor Set Request function and response processing
ZDO_ENDDEVICEBIND_REQUEST	Enable End Device Bind Request function and response processing
ZDO_BIND_UNBIND_REQUEST	Enable Bind and Unbind Request function and response processing
ZDO_SERVERDISC_REQUEST	Enable Server Discovery Request function and response processing
ZDO_MGMT_NWKDISC_REQUEST	Enable Mgmt Nwk Discovery Request function and response processing
ZDO_MGMT_LQI_REQUEST	Enable Mgmt LQI Request function and response processing
ZDO_MGMT_RTG_REQUEST	Enable Mgmt Routing Table Request function and response processing

ZDO_MGMT_BIND_REQUEST	Enable Mgmt Binding Table Request function and response processing
ZDO_MGMT_LEAVE_REQUEST	Enable Mgmt Leave Request function and response processing
ZDO_MGMT_JOINDIRECT_REQUEST	Enable Mgmt Join Direct Request function and response processing
ZDO_MGMT_PERMIT_JOIN_REQUEST	Enable device to respond to Mgmt Permit Join Request function
ZDO_ENDDEVICE_ANNCE_REQUEST	Enable device to respond to End Device Annce Request function
ZDO_USERDESC_RESPONSE	Enable device to respond to User Descriptor Request function
ZDO_USERDESCSET_RESPONSE	Enable device to respond to User Descriptor Set Request function
ZDO_SERVERDISC_RESPONSE	Enable device to respond to Server Discovery Request function
ZDO_MGMT_NWKDISC_RESPONSE	Enable device to respond to Mgmt Network Discovery Request function
ZDO_MGMT_LQI_RESPONSE	Enable device to respond to Mgmt LQI Request function
ZDO_MGMT_RTG_RESPONSE	Enable device to respond to Mgmt Routing Table Request function
ZDO_MGMT_BIND_RESPONSE	Enable device to respond to Mgmt Binding Table Request function
ZDO_MGMT_LEAVE_RESPONSE	Enable device to respond to Mgmt Leave Request function
ZDO_MGMT_JOINDIRECT_RESPONSE	Enable device to respond to Mgmt Join Direct Request function
ZDO_MGMT_PERMIT_JOIN_RESPONSE	Enable device to respond to Mgmt Permit Join Request function
ZDO_ENDDEVICE_ANNCE	Enable device to respond to End Device Annce Message function