www.qorvo.com
lpw.support@qorvo.com

gpMacDispatcher Reference Manual

API Description

Version 2.10.2.0
November 15, 2021

Contents

Chapter 1

Introduction

This document describes in a formal manner the API interface that can be used to control all the functionality of the MacDispatcher.

Abbreviations

- MAC Medium Access Control
- MAL MAC Abstraction Layer
- RF4CE Radio Frequency for Consumer Electronics

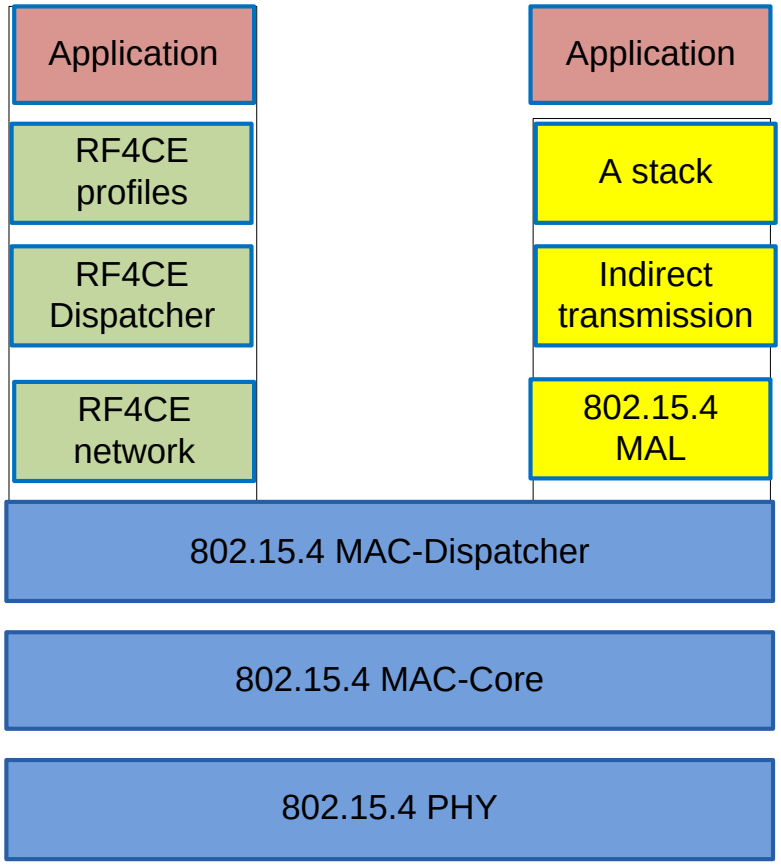

Figure 1.1: "MacDispatcher overview"

Chapter 2

Module Documentation

2.1 INIT primitives

Functions

- void gpMacDispatcher_Init (void)

  This function initializes the MacCore layer. It should be called before calling any other request function.
- gpMacDispatcher_Result_t gpMacDispatcher_Reset (Bool setDefaultPib, UInt8 stackId)

  This function resets the MACDispatcher + Core layer.
- gpMacDispatcher_StackId_t gpMacDispatcher_RegisterNetworkStack (gpMacDispatcher_StringIdentifier_t *stringIdentifier)

  This function is used to register a network stack to the dispatcher.

2.1.1 Detailed Description

This module groups the initialization routines.

2.1.2 Function Documentation

gpMacDispatcher_RegisterNetworkStack()

```
gpMacDispatcher_StackId_t gpMacDispatcher_RegisterNetworkStack (
              gpMacDispatcher_StringIdentifier_t * stringIdentifier )
```

Returns

The id that is allocated by the MacCore for this stack

gpMacDispatcher_Reset()

```
gpMacDispatcher_Result_t gpMacDispatcher_Reset (
              Bool setDefaultPib,
              UInt8 stackId )
```

Parameters

setDefaultPib	If true the PIB values are reset to their default value (as specified in the IEEE802.15.4-2006 specification).
stackId	The PIB values are reset to their default value (as specified in the IEEE802.15.4-2006 specification).

Note

A prerequisite; the invoker of this function should first aqcuire the Mac Dispatcher lock before a reset is allowed

Returns

Whether this stack has successfully issued a reset command on the MAC network layer access (true) or not (false).

2.2 DATA primitives

Functions

- void gpMacDispatcher_DataRequest (gpMacCore_AddressMode_t srcAddrMode, gpMacCore_AddressInfo_t *pDstAddrInfo, UInt8 txOptions, gpMacCore_Security_t *pSecOptions, gpMacCore_MultiChannelOptions_t multiChannelOptions, gpPd_Loh_t pdLoh, gpMacCore_StackId_t stackId)

  This function is used to send a data packet to another device.
- void gpMacDispatcher_PollRequest (gpMacCore_AddressInfo_t *pCoordAddrInfo, gpMacCore_Security_t *pSecOptions, gpMacCore_StackId_t stackId)

  This function is used to poll a coordinator for data by sending a data request command. If the coordinator has data pending for the device that issued the poll request, it will forward the data.
- void gpMacDispatcher_PurgeRequest (gpPd_Handle_t pdHandle, gpMacCore_StackId_t stackId)

  This function will remove the specified transmit request form the indirect transmission queue.

gpMacDispatcher_Result_t

- typedef void(* gpMacDispatcher_cbDataIndication_t) (const gpMacCore_AddressInfo_t *pSrcAddrInfo, const gpMacCore_AddressInfo_t *pDstAddrInfo, UInt8 dsn, gpMacCore_Security_t *pSecOptions, gpPd_Loh_t pdLoh, gpMacCore_StackId_t stackId)

  Calls the DataIndication callback function. It is used to inform the next higher layer a data packet is received.
- typedef void(* gpMacDispatcher_cbDataConfirm_t) (gpMacDispatcher_Result_t status, gpPd_Handle_t pdHandle, gpMacCore_StackId_t stackId)

  Calls the DataConfirm callback function. It is used to inform the next higher layer a data packet is sent.
- typedef void(* gpMacDispatcher_cbPollIndication_t) (gpMacCore_AddressInfo_t *pAddrInfo, gpPd_TimeStamp_t rxTime, gpMacCore_StackId_t stackId)

  Calls the PollIndication callback function. It is used to inform the next higher layer a Cmd DataRequest is received.
- typedef void(* gpMacDispatcher_cbPollConfirm_t) (gpMacDispatcher_Result_t status, gpMacCore_AddressInfo_t *pAddrInfo, gpPd_TimeStamp_t txTime, gpMacCore_StackId_t stackId)

  Calls the PollConfirm callback function. It is used to inform the next higher layer a poll request was sent (status).
- typedef void(* gpMacDispatcher_cbPurgeConfirm_t) (gpMacDispatcher_Result_t status, gpPd_Handle_t pdHandle, gpMacCore_StackId_t stackId)

  Calls the PurgeConfirm callback function. It is used to inform the next higher layer a purge request was performed (status).
- typedef void(* gpMacDispatcher_cbPollNotify_t) (gpMacCore_AddressInfo_t *pAddrInfo, gpPd_TimeStamp_t rxTime, gpPd_Handle_t pdHandle, Bool fromNeighbour, gpMacCore_StackId_t stackId)

  Calls the PollIndication callback function. It is used to inform the next higher layer a Cmd DataRequest is received.

2.2.1 Detailed Description

This module groups the primitives for data transmission.

- gpMacDispatcher_DataRequest + cbConfirm, cbIndication
- gpMacDispatcher_PollRequest + cbConfirm, cbIndication

2.2.2 Typedef Documentation

gpMacDispatcher_cbPollNotify_t

```
gpMacDispatcher_cbPollNotify_t
```

Parameters

pAddrInfo	The addressMode and address of PollRequest transmitter.
rxTime	The timestamp for the reception of the frame.
pdHandle	The pd Handle which will be transmitted as a response to the PollRequest. 0xFF indicates that no data is ready.
fromNeighbour	Boolean value, which indicates if the source address is present in our Neighbour list.

2.2.3 Function Documentation

gpMacDispatcher_DataRequest()

```
void gpMacDispatcher_DataRequest (
              gpMacCore_AddressMode_t srcAddrMode,
              gpMacCore_AddressInfo_t * pDstAddrInfo,
              UInt8 txOptions,
              gpMacCore_Security_t * pSecOptions,
              gpMacCore_MultiChannelOptions_t multiChannelOptions,
              gpPd_Loh_t pdLoh,
              gpMacCore_StackId_t stackId )
```

This function is used to send a data packet conform the IEEE802.15.4-2006 MAC specification.

Parameters

srcAddrMode	The address modes to be used for source address. This parameter is used as in the Frame Control Field of an IEEE802.15.4-2006 packet. We refer to the define GP_IEEEMAC_SRC_ADDR_(EXT/SHORT).
pDstAddrInfo	All the information about the destination (address mode, address and panId).
txOptions	Tx Options byte (see IEEE802.15.4-2006).
pSecOptions	The security options - maybe NULL to be used (see IEEE802.15.4-2006).
p_PdLoh	The PD Length/Offset/Handle of the data payload.
stackId	The stack doing this data request

gpMacDispatcher_PollRequest()

```
void gpMacDispatcher_PollRequest (
```

```
gpMacCore_AddressInfo_t * pCoordAddrInfo,
gpMacCore_Security_t * pSecOptions,
gpMacCore_StackId_t stackId )
```

This function starts a poll. This function is used to request packets that are pending at the coordinator (by sending a data request command to the coordinator). If the coordinator has data pending for the device that issued the poll request, it will forward the data. If a short address is allocated to the device, and it was written into the gpMacCore_AttributeShortAddress PIB attribute, the short address will be used as source address of the data request command. If the gpMacCore_AttributeShortAddress PIB attribute is set to 0xffff or 0xfffe, the device will use its extended address as source address in the data request command packet.

Parameters

pCoordAddrInfo	All the details regarding the coordinator
pSecOptions	structure with security options, if NULL no security is used.
stackId	The stack doing this poll request

2.3 MANAGEMENT primitives

Functions

- void gpMacDispatcher_ScanRequest (gpMacCore_ScanType_t scanType, UInt32 scanChannels, UInt8 scanDuration, UInt8 resultListSize, UInt8 *pResultList, gpMacCore_StackId_t stackId)

  This function is used to start a scan.
- void gpMacDispatcher_AssociateRequest (UInt8 logicalChannel, gpMacCore_AddressInfo_t *pCoordAddrInfo, UInt8 capabilityInformation, gpMacCore_StackId_t stackId)

  This function is issued by a device who wishes to join a network.
- void gpMacDispatcher_AssociateResponse (MACAddress_t *pDeviceAddress, UInt16 associateShortAddress, gpMacCore_Result_t status, gpMacCore_StackId_t stackId)

  This function is used to send an associate response command packet to a device that is trying to join. The associate response command packet contains the results of the join procedure.
- void gpMacDispatcher_OrphanResponse (MACAddress_t *pOrphanAddress, UInt16 shortAddress, Bool associatedMember, gpMacCore_StackId_t stackId)

  This function is used to send an orphan response command packet.

gpMacDispatcher_Result_t

- typedef void(* gpMacDispatcher_cbBeaconNotifyIndication_t) (UInt8 bsn, gpMacCore_PanDescriptor_t *pPanDescriptor, UInt8 beaconPayloadLength, UInt8 *pBeaconPayload, gpMacCore_StackId_t stackId)

  Calls the BeaconIndication callback function. It is used to inform the next higher layer a beacon was received.
- typedef void(* gpMacDispatcher_cbScanConfirm_t) (gpMacCore_Result_t status, gpMacCore_ScanType_t scanType, UInt32 unscannedChannels, UInt8 resultListSize, UInt8 *pResultList, gpMacCore_StackId_t stackId)

  Calls the ScanConfirm callback function. It is used to indicate the scan process was finished.
- typedef void(* gpMacDispatcher_cbAssocConfirm_t) (UInt16 assocShortAddress, gpMacCore_Result_t status, gpPd_TimeStamp_t txTime, gpMacCore_StackId_t stackId)

  Calls the AssocConfirm callback function. It is used to indicate the association process was finished.
- typedef void(* gpMacDispatcher_cbAssociateIndication_t) (gpMacCore_Address_t *pDeviceAddress, UInt8 capabilityInformation, gpPd_TimeStamp_t rxTime, gpMacCore_StackId_t stackId)

  Calls the AssocIndication callback function. It is used to indicate the association request was received.
- typedef void(* gpMacDispatcher_cbOrphanIndication_t) (MACAddress_t *pOrphanAddress, gpPd_TimeStamp_t rxTime, gpMacCore_StackId_t stackId)

  Calls the Orphan Indication callback function.
- typedef void(* gpMacDispatcher_cbSecurityFailureCommStatusIndication_t) (gpMacCore_AddressInfo_t *pSrcAddrInfo, gpMacCore_AddressInfo_t *pDstAddrInfo, gpMacCore_Result_t status, gpPd_TimeStamp_t txTime, gpMacCore_StackId_t stackId)

  Indicates a security failure communication status to the next higher layer.
- typedef void(* gpMacDispatcher_cbAssociateCommStatusIndication_t) (gpMacCore_AddressInfo_t *pSrcAddrInfo, gpMacCore_AddressInfo_t *pDstAddrInfo, gpMacCore_Result_t status, gpPd_TimeStamp_t txTime, gpMacCore_StackId_t stackId)

  Indicates a associate response communication status to the next higher layer.

- typedef void(∗ gpMacDispatcher_cbOrphanCommStatusIndication_t) (gpMacCore_AddressInfo_t ∗pSrcAddrInfo, gpMacCore_AddressInfo_t ∗pDstAddrInfo, gpMacCore_Result_t status, gpPd_TimeStamp_t txTime, gpMacCore_StackId_t stackId)

  Indicates an orphan communication status to the next higher layer.
- typedef void(∗ gpMacDispatcher_cbSecurityFrameCounterIndication_t) (UInt32 frameCounter, gpMacCore_StackId_t stackId)

  Indicates that the frame counter has receached a value that needs to be stored in NVM. The value should be restored by the stack upon restart by using the function gpMacDispatcher_SetFrameCounter();.

2.3.1 Detailed Description

- gpMacDispatcher_ScanRequest
- gpMacDispatcher_AssociateRequest
- gpMacDispatcher_AssociateResponse

2.3.2 Typedef Documentation

gpMacDispatcher_cbOrphanCommStatusIndication_t

`gpMacDispatcher_cbOrphanCommStatusIndication_t`

Indicates an unexpected driver reset to network stacks.

2.3.3 Function Documentation

gpMacDispatcher_AssociateRequest()

```
void gpMacDispatcher_AssociateRequest (
              UInt8 logicalChannel,
              gpMacCore_AddressInfo_t * pCoordAddrInfo,
              UInt8 capabilityInformation,
              gpMacCore_StackId_t stackId )
```

This function starts the associate procedure. It is used to join a network. When called, the AssociateRequest function will construct an associate request command packet, and send it to the coordinator.

Parameters

logicalChannel	The channel that will be used to do the association attempt
pCoordAddrInfo	The coordinator addressing info (address mode, panId, address)
capabilityInformation	Specifies the operational capabilities of the associating device.
stackId	The stack doing this associate request

gpMacDispatcher_AssociateResponse()

```
void gpMacDispatcher_AssociateResponse (
              MACAddress_t * pDeviceAddress,
              UInt16 associateShortAddress,
              gpMacCore_Result_t status,
              gpMacCore_StackId_t stackId )
```

This function is used to respond to a previously received associate request command packet. Upon reception of an associate request command packet, the AssociateIndication function is called to inform the higher layer. The next higher layer then decides to allow the joining device in the network or not. Its decision is sent back to the joining device through the associate response packet.

Parameters

pDeviceAddress	A pointer to the address of the device that is trying to join.
associateShortAddress	If the joining device is allowed (successful association), this parameter contains the short address that was allocated for the device. If the association was unsuccessful, this parameter is set to 0xffff.
status	This parameter contains the result of the association attempt. Possible values are: 0x0: Association successful. 0x1: PAN at capacity. 0x2: PAN access denied.
stackId	The stack doing this associate response

gpMacDispatcher_ScanRequest()

```
void gpMacDispatcher_ScanRequest (
              gpMacCore_ScanType_t scanType,
              UInt32 scanChannels,
              UInt8 scanDuration,
              UInt8 resultListSize,
              UInt8 * pResultList,
              gpMacCore_StackId_t stackId )
```

This function starts a scan conform the IEEE802.15.4-2006 spec. A scan can be issued to detect a channel with the least amount of interference, to find networks, etc.

Parameters

scanType	This parameter defines which of the 3 implemented scans needs to be performed: ED, active or orphan scan.
scanChannels	This parameter defines which channels need to be scanned. It is a bitmask where bit 0 must be '1' to select channel 0, bit 26 '1' to select channel 26, etc. Since the chip works in the 2.4GHz band, only channels 11 till 26 can be selected. If other channels are selected, they are ignored and only the selected channels between channel 11 and channel 26 are scanned.
scanDuration	This parameter specifies how long a channel will be scanned. The scantime equals [GP_IEEEMAC_BASE_SUPERFRAME_DURATION * (2^scanduration + 1)]symbols, where 1 symbol is 16us.

Parameters

resultListSize	The length of the resultList (in bytes). If an ED scan is done, this should equal the amount of channels. If resultListSize is higher than the amount of channels that needs to be scanned there is no problem. If the value is lower, the scan is terminated when the list is full. If an active scan is selected, resultListSize is the amount of PANDescriptors that can be saved. If it is an orphan scan, resultListSize is 0.
pResultList	This is a pointer to an array where the result of the scan can be saved. If an ED scan is done, the size is resultListSize. If an active scan is issued the size is resultListSize∗sizeof(gpMacCore_PanDescriptor_t).
stackId	The stack doing this scan request

2.4 ATTRIBUTE primitives

Macros

- #define gpMacDispatcher_SetAutoTxAntennaToggling(Bool, gpMacDispatcher_StackId_t)

  Enable or disable automatic toggling of TX antenna after transmission failure.

Functions

- void gpMacDispatcher_SetCurrentChannel (UInt8 channel, gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- void gpMacDispatcher_SetMultipleChannelTable (UInt8 *pChannel, gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- UInt8 gpMacDispatcher_GetCurrentChannel (gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- void gpMacDispatcher_SetDefaultTransmitPowers (Int8 *pDefaultTransmitPowerTable)

  Access method for the corresponding PIB attribute.
- void gpMacDispatcher_SetTransmitPower (gpMacCore_TxPower_t transmitPower, gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- Int8 gpMacDispatcher_GetTransmitPower (gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- void gpMacDispatcher_SetCCAMode (UInt8 cCAMode, gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- UInt8 gpMacDispatcher_GetCCAMode (gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- void gpMacDispatcher_SetCoordExtendedAddress (MACAddress_t *pCoordExtendedAddress, gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- void gpMacDispatcher_GetCoordExtendedAddress (MACAddress_t *pCoordExtendedAddress, gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- void gpMacDispatcher_SetCoordShortAddress (UInt16 addr, gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- UInt16 gpMacDispatcher_GetCoordShortAddress (gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- void gpMacDispatcher_SetPanCoordinator (Bool panCoordinator, gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- Bool gpMacDispatcher_GetPanCoordinator (gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- void gpMacDispatcher_SetDsn (UInt8 dsn, gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- UInt8 gpMacDispatcher_GetDsn (gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- void gpMacDispatcher_SetMaxCsmaBackoffs (UInt8 maxCsmaBackoffs, gpMacCore_StackId_t stackId)

Access method for the corresponding PIB attribute.

- UInt8 gpMacDispatcher_GetMaxCsmaBackoffs (gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- void gpMacDispatcher_SetMinBE (UInt8 minBE, gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- UInt8 gpMacDispatcher_GetMinBE (gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- void gpMacDispatcher_SetMaxBE (UInt8 maxBE, gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- UInt8 gpMacDispatcher_GetMaxBE (gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- void gpMacDispatcher_SetCsmaMode (UInt8 csmaMode, gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- UInt8 gpMacDispatcher_GetCsmaMode (gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- void gpMacDispatcher_SetPanId (UInt16 panId, gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- UInt16 gpMacDispatcher_GetPanId (gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- void gpMacDispatcher_SetRxOnWhenIdle (Bool rxOnWhenIdle, gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- Bool gpMacDispatcher_GetRxOnWhenIdle (gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- void gpMacDispatcher_SetShortAddress (UInt16 shortAddress, gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- UInt16 gpMacDispatcher_GetShortAddress (gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- void gpMacDispatcher_SetAssociationPermit (Bool associationPermit, gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- Bool gpMacDispatcher_GetAssociationPermit (gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- void gpMacDispatcher_SetBeaconPayload (UInt8 *pBeaconPayload, gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- void gpMacDispatcher_GetBeaconPayload (UInt8 *pBeaconPayload, gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- void gpMacDispatcher_SetBeaconPayloadLength (UInt8 length, gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- UInt8 gpMacDispatcher_GetBeaconPayloadLength (gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- void gpMacDispatcher_SetPromiscuousMode (UInt8 promiscuousMode, gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- UInt8 gpMacDispatcher_GetPromiscuousMode (gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.

- void gpMacDispatcher_SetTransactionPersistenceTime (UInt16 time, gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- UInt16 gpMacDispatcher_GetTransactionPersistenceTime (gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- void gpMacDispatcher_SetExtendedAddress (MACAddress_t *pExtendedAddress, gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- void gpMacDispatcher_GetExtendedAddress (MACAddress_t *pExtendedAddress, gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- void gpMacDispatcher_SetNumberOfRetries (UInt8 numberOfRetries, gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- UInt8 gpMacDispatcher_GetNumberOfRetries (gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- void gpMacDispatcher_SetSecurityEnabled (Bool securityEnabled, gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- Bool gpMacDispatcher_GetSecurityEnabled (gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- void gpMacDispatcher_SetBeaconStarted (Bool BeaconStarted, gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- Bool gpMacDispatcher_GetBeaconStarted (gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- void gpMacDispatcher_SetTxAntenna (UInt8 txAntenna, gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- UInt8 gpMacDispatcher_GetTxAntenna (gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- void gpMacDispatcher_SetIndicateBeaconNotifications (Bool enable, gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- Bool gpMacDispatcher_GetIndicateBeaconNotifications (gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- void gpMacDispatcher_SetForwardPollIndications (Bool enable, gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- Bool gpMacDispatcher_GetForwardPollIndications (gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- void gpMacDispatcher_SetLqiThresholdForTest (UInt8 lqiThreshold, gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- void gpMacDispatcher_SetAddressModeOverrideForBeacons (UInt8 addressMode, gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- void gpMacDispatcher_SetMacVersion (gpMacCore_MacVersion_t macVersion, gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- gpMacCore_MacVersion_t gpMacDispatcher_GetMacVersion (gpMacCore_StackId_t stackId)

Access method for the corresponding PIB attribute.

- gpMacCore_Result_t gpMacDispatcher_DataPending_QueueAdd (gpMacCore_AddressInfo_t *pAddrInfo, gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- gpMacCore_Result_t gpMacDispatcher_DataPending_QueueRemove (gpMacCore_AddressInfo_t *pAddrInfo, gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- gpMacCore_Result_t gpMacDispatcher_DataPending_QueueClear (gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- Bool gpMacDispatcher_AddNeighbour (gpMacCore_AddressInfo_t *pAddrInfo, gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- Bool gpMacDispatcher_RemoveNeighbour (gpMacCore_AddressInfo_t *pAddrInfo, gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- void gpMacDispatcher_ClearNeighbours (gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- gpMacCore_Result_t gpMacDispatcher_SetDataPendingMode (gpMacCore_DataPendingMode_t dataPendingMode, gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- void gpMacDispatcher_SetFrameCounter (UInt32 frameCounter, gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- UInt32 gpMacDispatcher_GetFrameCounter (gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- gpMacCore_Result_t gpMacDispatcher_SetKeyDescriptor (gpMacCore_KeyDescriptor_t *pKeyDescriptor, gpMacCore_Index_t index, gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- gpMacCore_Result_t gpMacDispatcher_GetKeyDescriptor (gpMacCore_KeyDescriptor_t *pKeyDescriptor, gpMacCore_Index_t index, gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- void gpMacDispatcher_SetKeyTableEntries (gpMacCore_KeyTablesEntries_t keyTableEntries, gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- gpMacCore_KeyTablesEntries_t gpMacDispatcher_GetKeyTableEntries (gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- gpMacCore_Result_t gpMacDispatcher_SetDeviceDescriptor (gpMacCore_DeviceDescriptor_t *pDeviceDescriptor, gpMacCore_Index_t index, gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- gpMacCore_Result_t gpMacDispatcher_GetDeviceDescriptor (gpMacCore_DeviceDescriptor_t *pDeviceDescriptor, gpMacCore_Index_t index, gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- void gpMacDispatcher_SetDeviceTableEntries (gpMacCore_DeviceTablesEntries_t deviceTableEntries, gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- gpMacCore_DeviceTablesEntries_t gpMacDispatcher_GetDeviceTableEntries (gpMacCore_StackId_t stackId)

Access method for the corresponding PIB attribute.

- gpMacCore_Result_t gpMacDispatcher_SetSecurityLevelDescriptor (gpMacCore_SecurityLevelDescriptor_t *pSecurityLevelDescriptor, gpMacCore_Index_t index, gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- gpMacCore_Result_t gpMacDispatcher_GetSecurityLevelDescriptor (gpMacCore_SecurityLevelDescriptor_t *pSecurityLevelDescriptor, gpMacCore_Index_t index, gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- gpMacCore_SecurityLevelTableEntries_t gpMacDispatcher_GetSecurityLevelTableEntries (gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- void gpMacDispatcher_SetSecurityLevelTableEntries (gpMacCore_SecurityLevelTableEntries_t securityLevelTableEntries, gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- void gpMacDispatcher_SetDefaultKeySource (UInt8 *pDefaultKeySource, gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- void gpMacDispatcher_GetDefaultKeySource (UInt8 *pDefaultKeySource, gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- void gpMacDispatcher_SetPanCoordExtendedAddress (MACAddress_t *pPanCoordExtendedAddress, gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- void gpMacDispatcher_GetPanCoordExtendedAddress (MACAddress_t *pPanCoordExtendedAddress, gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- void gpMacDispatcher_SetPanCoordShortAddress (UInt16 PanCoordShortAddress, gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- UInt16 gpMacDispatcher_GetPanCoordShortAddress (gpMacCore_StackId_t stackId)

  Access method for the corresponding PIB attribute.
- Bool gpMacDispatcher_GetAutoTxAntennaToggling (gpMacDispatcher_StackId_t stackId)

  Return the current state of automatic toggling of TX antenna.

2.4.1 Detailed Description

- gpMacDispatcher_ScanRequest
- gpMacDispatcher_AssociateRequest
- gpMacDispatcher_AssociateResponse

2.4.2 Macro Definition Documentation

gpMacDispatcher_SetAutoTxAntennaToggling

```
#define gpMacDispatcher_SetAutoTxAntennaToggling(
                Bool,
                gpMacDispatcher_StackId_t )
```

Parameters

enable	True to enable auto-toggling, false to disable.
stackId	Stack identifier.

2.4.3 Function Documentation

gpMacDispatcher_GetAutoTxAntennaToggling()

```
Bool gpMacDispatcher_GetAutoTxAntennaToggling (
              gpMacDispatcher_StackId_t stackId )
```

Parameters

stackId	Stack identifier.

Returns

true when enabled, false when disabled.

Chapter 3

Data Structure Documentation

3.1 gpMacDispatcher_Callbacks_t Struct Reference

Data Fields

- gpMacDispatcher_cbDataIndication_t **dataIndicationCallback**
- gpMacDispatcher_cbDataConfirm_t **dataConfirmCallback**
- gpMacDispatcher_cbPollIndication_t **pollIndicationCallback**
- gpMacDispatcher_cbPollConfirm_t **pollConfirmCallback**
- gpMacDispatcher_cbPurgeConfirm_t **purgeConfirmCallback**
- gpMacDispatcher_cbBeaconNotifyIndication_t **beaconNotifyIndicationCallback**
- gpMacDispatcher_cbScanConfirm_t **scanConfirmCallback**
- gpMacDispatcher_cbAssociateIndication_t **assocIndicationCallback**
- gpMacDispatcher_cbAssocConfirm_t **assocConfirmCallback**
- gpMacDispatcher_cbOrphanIndication_t **orphanIndicationCallback**
- gpMacDispatcher_cbSecurityFailureCommStatusIndication_t **securityFailureCommStatusIndicationCallback**
- gpMacDispatcher_cbAssociateCommStatusIndication_t **associateCommStatusIndicationCallback**
- gpMacDispatcher_cbOrphanCommStatusIndication_t **orphanCommStatusIndicationCallback**
- gpMacDispatcher_cbDriverResetIndication_t **driverResetIndicationCallback**
- gpMacDispatcher_cbPollNotify_t **pollNotifyCallback**
- gpMacDispatcher_cbSecurityFrameCounterIndication_t **securityFrameCounterIndicationCallback**

3.2 gpMacDispatcher_StringIdentifier_t Struct Reference

Data Fields

- char **str** [GP_MAC_DISPATCHER_STRING_ID_LENGTH]

Chapter 4

File Documentation

4.1 gpMacDispatcher.h File Reference

The MacDispatcher locks the lower layer before processing any request, after the cbConfirm the lock is released.

Data Structures

- struct gpMacDispatcher_StringIdentifier_t
- struct gpMacDispatcher_Callbacks_t

Macros

- #define GP_MAC_DISPATCHER_INVALID_STACK_ID (GP_MACCORE_STACK_UNDEFINED)

  The invalid stack ID.
- #define **GP_MAC_DISPATCHER_STRING_ID_LENGTH** 5

gpMacDispatcher_Result_t

- #define gpMacDispatcher_ResultSuccess gpMacCore_ResultSuccess

  The requested operation was completed successfully. For instance if a transmission was requested, this value indicates a successful transmission.
- #define gpMacDispatcher_ResultCounterError gpMacCore_ResultCounterError

  The frame counter purportedly applied by the originator of the received frame is invalid.
- #define gpMacDispatcher_ResultImproperKeyType gpMacCore_ResultImproperKeyType

  The key purportedly applied by the originator of the received frame is not allowed to be used with that frame type according to the key usage policy of the recipient.
- #define gpMacDispatcher_ResultImproperSecurityLevel gpMacCore_ResultImproperSecurityLevel

  The security level purportedly applied by the originator of the received frame does not meet the minimum security level required/expected by the recipient for that frame type.
- #define gpMacDispatcher_ResultUnsupportedLegacy gpMacCore_ResultUnsupportedLegacy

  The received frame was purportedly secured using security based on IEEE Std 802.15.4-2003, and such security is not supported by this standard.
- #define gpMacDispatcher_ResultUnsupportedSecurity gpMacCore_ResultUnsupportedSecurity

  The security purportedly applied by the originator of the received frame is not supported.
- #define gpMacDispatcher_ResultBeaconLoss gpMacCore_ResultBeaconLoss

  The beacon was lost following a synchronization request.

- #define gpMacDispatcher_ResultChannelAccessFailure gpMacCore_ResultChannelAccessFailure

  A transmission could not take place due to activity on the channel, i.e., the CSMA-CA mechanism has failed.
- #define gpMacDispatcher_ResultDenied gpMacCore_ResultDenied

  The GTS request has been denied by the PAN coordinator.
- #define gpMacDispatcher_ResultDisableTrxFailure gpMacCore_ResultDisableTrxFailure

  The attempt to disable the transceiver has failed.
- #define gpMacDispatcher_ResultSecurityError gpMacCore_ResultSecurityError

  The received frame induces a failed security check according to the security suite.
- #define gpMacDispatcher_ResultFrameTooLong gpMacCore_ResultFrameTooLong

  The frame resulting from secure processing has a length that is greater than aMACMaxFrameSize.
- #define gpMacDispatcher_ResultInvalidGTS gpMacCore_ResultInvalidGTS

  The requested GTS transmission failed because the specified GTS either did not have a transmit GTS direction or was not defined.
- #define gpMacDispatcher_ResultInvalidHandle gpMacCore_ResultInvalidHandle

  A request to purge an MSDU from the transaction queue wasmade using an MSDU handle that was not found in the transaction table.
- #define gpMacDispatcher_ResultInvalidParameter gpMacCore_ResultInvalidParameter

  A parameter in the primitive is out of the valid range.
- #define gpMacDispatcher_ResultNoAck gpMacCore_ResultNoAck

  No acknowledgment was received after aMaxFrameRetries.
- #define gpMacDispatcher_ResultNoBeacon gpMacCore_ResultNoBeacon

  A scan operation failed to find any network beacons.
- #define gpMacDispatcher_ResultNoData gpMacCore_ResultNoData

  No response data was available following a request.
- #define gpMacDispatcher_ResultNoShortAddress gpMacCore_ResultNoShortAddress

  The operation failed because a short address was not allocated.
- #define gpMacDispatcher_ResultOutOfCAP gpMacCore_ResultOutOfCAP

  A request to enable the receiver was unsuccessful because it could not be completed within the CAP.
- #define gpMacDispatcher_ResultPanIdConflict gpMacCore_ResultPanIdConflict

  A PAN identifier conflict has been detected and communicated to the PAN coordinator.
- #define gpMacDispatcher_ResultRealignment gpMacCore_ResultRealignment

  A coordinator realignment command has been received.
- #define gpMacDispatcher_ResultTransactionExpired gpMacCore_ResultTransactionExpired

  The transaction has expired and its information is discarded.
- #define gpMacDispatcher_ResultTransactionOverflow gpMacCore_ResultTransactionOverflow

  There is no capacity to store the transaction.
- #define gpMacDispatcher_ResultTxActive gpMacCore_ResultTxActive

  The transceiver was transmitting when the receiver was requested to be enabled.
- #define gpMacDispatcher_ResultUnavailableKey gpMacCore_ResultUnavailableKey

  The appropriate key is not available in the ACL.
- #define gpMacDispatcher_ResultUnsupportedAttribute gpMacCore_ResultUnsupportedAttribute

  A SET/GET request was issued with the identifier of a PIBattribute that is not supported.
- #define gpMacDispatcher_ResultInvalidAddress gpMacCore_ResultInvalidAddress

  A request to send data was unsuccessful because neither the source address parameters nor the destination address parameters were present.
- #define gpMacDispatcher_ResultOnTimeTooLong gpMacCore_ResultOnTimeTooLong

A receiver enable request was unsuccessful because it specified a number of symbols that was longer than the beacon interval.

- #define gpMacDispatcher_ResultPastTime gpMacCore_ResultPastTime

  A receiver enable request was unsuccessful because it could not be completed within the current superframe and was not permitted to be deferred until the next superframe.
- #define gpMacDispatcher_ResultTrackingOff gpMacCore_ResultTrackingOff

  The device was instructed to start sending beacons based on the timing of the beacon transmissions of its coordinator, but the device is not currently tracking the beacon of its coordinator.
- #define gpMacDispatcher_ResultInvalidIndex gpMacCore_ResultInvalidIndex

  An attempt to write to a MAC PIB attribute that is in a table failed because the specified table index was out of range.
- #define gpMacDispatcher_ResultLimitedReached gpMacCore_ResultLimitedReached

  A scan operation terminated prematurely because the number of PAN descriptors stored reached an implementationspecified maximum.
- #define gpMacDispatcher_ResultReadOnly gpMacCore_ResultReadOnly

  A SET/GET request was issued with the identifier of an attribute that is read only.
- #define gpMacDispatcher_ResultScanInProgress gpMacCore_ResultScanInProgress

  A request to perform a scan operation failed because the MLME was in the process of performing a previously initiated scan operation.
- #define gpMacDispatcher_ResultSuperframeOverlap gpMacCore_ResultSuperframeOverlap

  The device was instructed to start sending beacons based on the timing of the beacon transmissions of its coordinator, but the instructed start time overlapped the transmission time of the beacon of its coordinator.
- #define gpMacDispatcher_ResultSecondStackRegistered 0xFE

  Status send on driver reset indication to indicate that a second stack with the same ID has been registered.
- #define gpMacDispatcher_ResultResetFinished 0xFF

  Status send on driver reset indication to indicate that a reset has finished - used for server/client only.
- typedef gpMacCore_Result_t **gpMacDispatcher_Result_t**
- typedef UInt8 **gpMacDispatcher_StackId_t**
- typedef void(∗ gpMacDispatcher_cbDataIndication_t) (const gpMacCore_AddressInfo_t ∗pSrcAddrInfo, const gpMacCore_AddressInfo_t ∗pDstAddrInfo, UInt8 dsn, gpMacCore_Security_t ∗pSecOptions, gpPd_Loh_t pdLoh, gpMacCore_StackId_t stackId)

  Calls the DataIndication callback function. It is used to inform the next higher layer a data packet is received.
- typedef void(∗ gpMacDispatcher_cbDataConfirm_t) (gpMacDispatcher_Result_t status, gpPd_Handle_t pdHandle, gpMacCore_StackId_t stackId)

  Calls the DataConfirm callback function. It is used to inform the next higher layer a data packet is sent.
- typedef void(∗ gpMacDispatcher_cbPollIndication_t) (gpMacCore_AddressInfo_t ∗pAddrInfo, gpPd_TimeStamp_t rxTime, gpMacCore_StackId_t stackId)

  Calls the PollIndication callback function. It is used to inform the next higher layer a Cmd DataRequest is received.
- typedef void(∗ gpMacDispatcher_cbPollConfirm_t) (gpMacDispatcher_Result_t status, gpMacCore_AddressInfo_t ∗pAddrInfo, gpPd_TimeStamp_t txTime, gpMacCore_StackId_t stackId)

  Calls the PollConfirm callback function. It is used to inform the next higher layer a poll request was sent (status).
- typedef void(∗ gpMacDispatcher_cbPurgeConfirm_t) (gpMacDispatcher_Result_t status, gpPd_Handle_t pdHandle, gpMacCore_StackId_t stackId)

Calls the PurgeConfirm callback function. It is used to inform the next higher layer a purge request was performed (status).

- typedef void(∗ gpMacDispatcher_cbBeaconNotifyIndication_t) (UInt8 bsn, gpMacCore_PanDescriptor_t ∗pPanDescriptor, UInt8 beaconPayloadLength, UInt8 ∗pBeaconPayload, gpMacCore_StackId_t stackId)

  Calls the BeaconIndication callback function. It is used to inform the next higher layer a beacon was received.
- typedef void(∗ gpMacDispatcher_cbScanConfirm_t) (gpMacCore_Result_t status, gpMacCore_ScanType_t scanType, UInt32 unscannedChannels, UInt8 resultListSize, UInt8 ∗pResultList, gpMacCore_StackId_t stackId)

  Calls the ScanConfirm callback function. It is used to indicate the scan process was finished.
- typedef void(∗ gpMacDispatcher_cbAssocConfirm_t) (UInt16 assocShortAddress, gpMacCore_Result_t status, gpPd_TimeStamp_t txTime, gpMacCore_StackId_t stackId)

  Calls the AssocConfirm callback function. It is used to indicate the association process was finished.
- typedef void(∗ gpMacDispatcher_cbAssociateIndication_t) (gpMacCore_Address_t ∗pDeviceAddress, UInt8 capabilityInformation, gpPd_TimeStamp_t rxTime, gpMacCore_StackId_t stackId)

  Calls the AssocIndication callback function. It is used to indicate the association request was received.
- typedef void(∗ gpMacDispatcher_cbOrphanIndication_t) (MACAddress_t ∗pOrphanAddress, gpPd_TimeStamp_t rxTime, gpMacCore_StackId_t stackId)

  Calls the Orphan Indication callback function.
- typedef void(∗ gpMacDispatcher_cbSecurityFailureCommStatusIndication_t) (gpMacCore_AddressInfo_t ∗pSrcAddrInfo, gpMacCore_AddressInfo_t ∗pDstAddrInfo, gpMacCore_Result_t status, gpPd_TimeStamp_t txTime, gpMacCore_StackId_t stackId)

  Indicates a security failure communication status to the next higher layer.
- typedef void(∗ gpMacDispatcher_cbAssociateCommStatusIndication_t) (gpMacCore_AddressInfo_t ∗pSrcAddrInfo, gpMacCore_AddressInfo_t ∗pDstAddrInfo, gpMacCore_Result_t status, gpPd_TimeStamp_t txTime, gpMacCore_StackId_t stackId)

  Indicates a associate response communication status to the next higher layer.
- typedef void(∗ gpMacDispatcher_cbOrphanCommStatusIndication_t) (gpMacCore_AddressInfo_t ∗pSrcAddrInfo, gpMacCore_AddressInfo_t ∗pDstAddrInfo, gpMacCore_Result_t status, gpPd_TimeStamp_t txTime, gpMacCore_StackId_t stackId)

  Indicates an orphan communication status to the next higher layer.
- typedef void(∗ gpMacDispatcher_cbPollNotify_t) (gpMacCore_AddressInfo_t ∗pAddrInfo, gpPd_TimeStamp_t rxTime, gpPd_Handle_t pdHandle, Bool fromNeighbour, gpMacCore_StackId_t stackId)

  Calls the PollIndication callback function. It is used to inform the next higher layer a Cmd DataRequest is received.
- typedef void(∗ **gpMacDispatcher_cbDriverResetIndication_t**) (gpMacCore_Result_t status, gpMacCore_StackId_t stackId)
- typedef void(∗ gpMacDispatcher_cbSecurityFrameCounterIndication_t) (UInt32 frameCounter, gpMacCore_StackId_t stackId)

  Indicates that the frame counter has receached a value that needs to be stored in NVM. The value should be restored by the stack upon restart by using the function gpMacDispatcher_SetFrameCounter();.
- void gpMacDispatcher_RegisterCallbacks (gpMacDispatcher_StackId_t stackId, gpMacDispatcher_Callbacks_t ∗pCallbacks)

  This function is used to set the callbacks for a specific stack.
- void **gpMacDispatcher_GetCallbacks** (gpMacDispatcher_StackId_t stackId, gpMacDispatcher_Callbacks_t ∗pCallbacks)
- void **gpMacDispatcher_SetMinInterferenceLevels** (Int8 ∗pInterferenceLevels)